



Guida per l'utente

Amazon ElastiCache



Versione API 2015-02-02

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

Amazon ElastiCache: 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?	1
Cache serverless	1
Cluster progettati autonomamente	2
Servizi correlati	2
Come funziona	3
Motori di cache e caching	3
Scelta tra le opzioni di implementazione	9
ElastiCache risorse	16
AWS Regioni e zone di disponibilità	18
Casi d'uso	19
Iniziare con ElastiCache	27
Configurazione ElastiCache	27
Registrati per un Account AWS	27
Crea un utente con accesso amministrativo	28
Concessione dell'accesso programmatico	29
Impostazione delle autorizzazioni	31
Configurare EC2	32
Assegnazione dell'accesso di rete	33
Configura l'accesso alla riga di comando	34
Crea una cache senza server Valkey	35
Lettura e scrittura di dati	36
Eliminazione	38
Fasi successive	39
Crea una cache serverless Valkey o Redis OSS	39
Lettura e scrittura di dati	40
Eliminazione	42
Fasi successive	43
Crea una cache serverless Memcached	43
Lettura e scrittura di dati	44
Eliminazione	49
Fasi successive	50
Tutorial: Guida introduttiva a Python e ElastiCache	50
Python e ElastiCache	51
Tutorial: Configurazione di Lambda per ElastiCache l'accesso in un VPC	69

Passaggio 1: creazione di una cache serverless. ElastiCache	69
Fase 2: Creare una funzione Lambda per ElastiCache	72
Fase 3: testare la funzione Lambda con ElastiCache	76
Fase 4: Pulizia (opzionale)	76
Progettazione del proprio ElastiCache cluster	78
Componenti e caratteristiche	78
Nodi	79
ElastiCache frammenti	79
ElastiCache cluster	80
ElastiCache replica	82
ElastiCache endpoint	84
Gruppi di parametri	85
ElastiCache sicurezza	86
Gruppi di sottoreti	86
ElastiCache backup	87
Eventi	87
ElastiCache terminologia	88
Tutorial: Come progettare il proprio cluster	90
Progettazione del proprio ElastiCache cluster (Valkey)	90
Progettazione del proprio ElastiCache cluster (RedisOSS)	111
Eliminazione di un cluster	132
Altri tutorial e video	133
Video	135
Gestione dei nodi in ElastiCache	140
Visualizzazione dello stato ElastiCache del nodo	141
OSSNodi e shard Valkey o Redis	148
Connessione ai nodi	150
Tipi di nodi supportati	155
Riavvio dei nodi	170
Sostituzione dei nodi (Valkey e RedisOSS)	175
Sostituzione dei nodi (Memcached)	182
Nodi riservati	184
Migrazione dei nodi della generazione precedente	200
Gestione dei cluster in ElastiCache	203
Scelta del tipo di rete in ElastiCache	206
Discovery automatico (Memcached)	211

Suddivisione dei dati su più livelli in ElastiCache	254
Preparazione di un cluster in ElastiCache	261
Creazione di un cluster per Valkey o Redis OSS	271
Creazione di un cluster per Memcached	281
Visualizzazione dei dettagli di un ElastiCache cluster	285
Modifica di un cluster ElastiCache	299
Aggiungere nodi a un ElastiCache cluster	305
Rimozione di nodi da un ElastiCache cluster	316
Annullamento delle operazioni di aggiunta o eliminazione dei nodi in sospeso in ElastiCache	325
Eliminazione di un cluster in ElastiCache	326
Accesso al ElastiCache cluster o al gruppo di replica	329
Ricerca degli endpoint di connessione in ElastiCache	338
Frammenti dentro ElastiCache	353
Confronto tra le cache autoprogettate da ValkeyOSS, Redis e Memcached	359
Migrazione online per Valkey o Redis OSS	364
Panoramica	364
Fasi della migrazione	365
Preparazione dell'origine e della destinazione per la migrazione	365
Test della migrazione dei dati	367
Avvio della migrazione	367
Verifica dell'avanzamento della migrazione dei dati	369
Completamento della migrazione dei dati	370
Esecuzione della migrazione dei dati online utilizzando la console	371
Scelta delle regioni e delle zone di disponibilità per ElastiCache	372
Considerazioni sulla zona di disponibilità con Memcached	373
Ubicazione dei nodi	376
Regioni ed endpoint supportati	376
Utilizzo delle zone locali con ElastiCache	381
Usare Outposts con ElastiCache	383
Lavorare con ElastiCache	387
Snapshot e ripristino	387
Vincoli	388
Impatto sulle prestazioni dei backup di cluster progettati autonomamente	389
Pianificazione di backup automatici	391
Esecuzione di backup manuali	393

Creazione di un backup finale	399
Descrizione dei backup	402
Copia di backup	404
Esportazione di un backup	406
Ripristino da un backup	414
Eliminazione di un backup	417
Assegnazione di tag di backup	418
Tutorial: Eseguire il seeding di un cluster progettato autonomamente con un backup	420
Versioni del motore e aggiornamento in ElastiCache	430
Gestione delle versioni per ElastiCache	430
Come aggiornare la versione di un motore	435
Come attivare l'aggiornamento cross-engine da Redis OSS a Valkey	435
Versioni supportate	437
Principali differenze di comportamento e compatibilità della versione con Valkey	459
Principali differenze di comportamento e compatibilità delle versioni con Redis OSS	459
Risoluzione degli aggiornamenti del motore bloccati con Valkey o Redis OSS	463
Best practice e strategie di caching	464
Le migliori pratiche generali	464
Comandi Valkey, Redis e Memcached supportati OSS e limitati	466
OSSConfigurazione e limiti di Valkey e Redis	502
IPv6esempi di client per Valkey, Redis OSS e Memcached	506
Le migliori pratiche per i clienti (Valkey e RedisOSS)	507
Le migliori pratiche per i clienti (Memcached)	530
TLSCluster dual stack ElastiCache abilitati	534
Gestione della memoria riservata per Valkey e Redis OSS	537
Le migliori pratiche per lavorare con cluster progettati autonomamente da Valkey e Redis OSS	544
Strategie di caching per Memcached	550
Gestione del cluster progettato autonomamente in ElastiCache	555
Auto Scaling dei cluster Valkey e Redis OSS	556
Modifica della modalità cluster	604
Replica tra AWS regioni utilizzando datastore globali	608
Alta disponibilità utilizzando gruppi di replica	636
Gestione della manutenzione dei ElastiCache cluster	726
Configurazione dei parametri del motore utilizzando gruppi di ElastiCache parametri	729
Ridimensionamento ElastiCache	839

ElastiCache Scalabilità senza server	839
Impostazione dei limiti di dimensionamento per gestire i costi	840
Prescalabilità con Serverless ElastiCache	840
Impostazione dei limiti di scalabilità utilizzando la console e AWS CLI	842
Scalabilità dei cluster progettati autonomamente	843
Guida introduttiva JSON a Valkey e Redis OSS	921
JSON panoramica dei tipi di dati	922
JSON comandi	934
Taggare le tue risorse ElastiCache	976
Monitoraggio dei costi con i tag	987
Gestione dei tag utilizzando il AWS CLI	989
Gestione dei tag utilizzando il ElastiCache API	994
Obiettivo Amazon ElastiCache Well-Architected	997
Pilastro dell'eccellenza operativa	998
Pilastro della sicurezza	1006
Pilastro dell'affidabilità	1013
Pilastro dell'efficienza delle prestazioni	1019
Pilastro dell'ottimizzazione dei costi	1030
Risoluzione dei problemi in ElastiCache	1036
Problemi di connessione	1037
Errori del client Valkey o Redis OSS	1037
Risoluzione dei problemi di latenza elevata in Serverless ElastiCache	1038
Risoluzione dei problemi di throttling in Serverless ElastiCache	1040
Problemi di connessione persistenti	1041
Argomenti correlati	1062
Sicurezza	1063
Protezione dei dati	1064
Sicurezza dei dati in Amazon ElastiCache	1064
Riservatezza del traffico Internet	1142
Amazon VPCs e la ElastiCache sicurezza	1142
ElastiCache API e VPC endpoint di interfaccia ()AWS PrivateLink	1168
Sottoreti e gruppi di sottoreti	1171
Identity and Access Management	1180
Destinatari	1180
Autenticazione con identità	1181
Gestione dell'accesso con policy	1184

Come ElastiCache funziona Amazon con IAM	1187
Esempi di policy basate su identità	1194
Risoluzione dei problemi	1197
Controllo accessi	1199
Panoramica sulla gestione degli accessi	1200
Convalida della conformità	1247
Ulteriori informazioni	1249
Resilienza	1249
Limitazione dell'impatto degli errori	1249
Sicurezza dell'infrastruttura	1254
Aggiornamenti di servizio	1255
Gestione degli aggiornamenti del servizio	1255
Vulnerabilità di sicurezza risolte	1261
Registrazione di log e monitoraggio	1263
Metriche ed eventi serverless per Valkey e Redis OSS	1263
Metriche serverless	1263
Eventi serverless	1272
Metriche ed eventi del cluster progettati autonomamente	1287
Metriche ed eventi serverless per Memcached	1296
Metriche serverless	1296
Eventi serverless	1300
Registrazione delle ElastiCache API chiamate Amazon con AWS CloudTrail	1311
ElastiCache Informazioni su Amazon in CloudTrail	1312
Comprendere le voci dei file di ElastiCache log di Amazon	1313
Monitoraggio SNS degli eventi Amazon	1316
Gestione delle SNS notifiche ElastiCache Amazon	1317
Visualizzazione ElastiCache degli eventi	1322
Notifiche di eventi e Amazon SNS	1326
Consegna dei registri	1334
Contenuto di una voce di registro lenta	1335
Contenuto di una voce del registro motore	1335
Autorizzazioni per configurare la registrazione	1335
Specificazioni del tipo di registro e del formato	1336
ElastiCache destinazioni di registrazione	1337
Specifica della consegna del registro tramite la console	1340
Specificare la consegna dei registri utilizzando il AWS CLI	1341

Monitoraggio dell'utilizzo	1346
Parametri a livello di host	1347
Metriche per Valkey e Redis OSS	1351
Parametri per Memcached	1369
Quali parametri è opportuno monitorare?	1374
Scelta delle statistiche e dei periodi di un parametro	1378
Monitoraggio delle metriche dei CloudWatch cluster e dei nodi	1378
Quote	1382
Riferimento	1384
Usando il ElastiCache API	1384
Utilizzo dell'interrogazione API	1384
Librerie disponibili	1388
Risoluzione dei problemi delle applicazioni	1388
Configura il AWS CLI per ElastiCache	1389
Prerequisiti	1390
Ottenere gli strumenti a riga di comando	1391
Configurazione degli strumenti	1392
Fornire le credenziali per gli strumenti	1393
Variabili di ambiente	1394
Messaggi di errore	1395
Notifiche	1396
ElastiCache Notifiche generali	1397
ElastiCache Notifiche (Memcached)	1397
ElastiCache (RedisOSS) notifiche specifiche	1397
ElastiCache Cronologia della documentazione	1398
AWS Glossario	1438
.....	mcdxxxix

Che cos'è Amazon ElastiCache?

Benvenuto nella Amazon ElastiCache User Guide. 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 i motori Valkey, Redis e OSS Memcached. Se non sei certo di quale motore utilizzare, consulta [Confronto tra le cache autoprogettate da ValkeyOSS, Redis e Memcached](#) in questa guida.

Cache serverless

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

ElastiCache Serverless elimina inoltre la necessità di pianificare e gestire la capacità di memorizzazione nella cache. ElastiCache monitora costantemente la memoria della cache, l'elaborazione e la larghezza di banda di rete utilizzata dall'applicazione e ridimensiona per soddisfare le esigenze dell'applicazione. ElastiCache offre un'esperienza endpoint semplice per gli sviluppatori, astruendo l'infrastruttura di cache sottostante e la progettazione del cluster. ElastiCache gestisce il provisioning dell'hardware, il monitoraggio, la sostituzione dei nodi e l'applicazione di patch software in modo automatico e trasparente, in modo da potersi concentrare sullo sviluppo delle applicazioni anziché sull'utilizzo della cache.

ElastiCache Serverless è compatibile con Valkey 7.2, Redis OSS 7.1 e versioni successive e Memcached 1.6.21 e versioni successive.

ElastiCache Progettazione del proprio cluster

Se hai bisogno di un controllo granulare sul tuo ElastiCache cluster, puoi scegliere di progettare il tuo cluster Valkey, Redis OSS o Memcached 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 del proprio ElastiCache cluster 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 Valkey, Redis OSS o Memcached in modalità cluster, abilitando la scalabilità orizzontale, oppure senza la modalità cluster per la semplice scalabilità 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. Puoi anche scegliere quando applicare nuove patch software al tuo cluster Valkey o Redis. OSS

Quando si progetta il proprio ElastiCache cluster, è possibile scegliere di eseguire Valkey 7.2, Redis da OSS 4.0 a 7.1 o Memcached 1.4 e versioni successive.

Servizi correlati

[DB di memoria](#)

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

- ElastiCache è un servizio che viene comunemente utilizzato per memorizzare nella cache i dati da altri database e archivi dati utilizzando Valkey, Redis o Memcached. OSS È consigliabile prendere in considerazione ElastiCache la possibilità di memorizzare nella cache i carichi di lavoro in cui si desidera accelerare l'accesso ai dati con il database o l'archivio dati primario esistente (prestazioni di lettura e scrittura in microsecondi). È inoltre necessario prendere in considerazione ElastiCache i casi d'uso in cui si desidera utilizzare strutture di dati Valkey o Redis e accedere APIs ai OSS dati archiviati in un database o in un data store primario.
- ElastiCache può anche aiutarvi a risparmiare sui costi del database archiviando i dati a cui si accede di frequente in una cache. Se l'applicazione ha requisiti di velocità effettiva di lettura elevati, è possibile ottenere scalabilità elevata, prestazioni veloci e costi di archiviazione dei dati ridotti utilizzando ElastiCache, anziché scalare, il database sottostante.

- MemoryDB è un database durevole in memoria per carichi di lavoro che richiedono un database primario ultraveloce. È compatibile con Valkey e Redis. OSS 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 tuo caso d'uso se desideri creare un'applicazione utilizzando strutture di OSS dati Valkey o Redis e APIs con un database primario e 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 Relational Database Service](#)

ElastiCache può aiutarvi a risparmiare sui costi del database archiviando i dati a cui si accede di frequente in una cache. Se l'applicazione ha requisiti di throughput di lettura elevati, è possibile ottenere scalabilità elevata, prestazioni veloci e costi di storage dei dati ridotti utilizzando ElastiCache, anziché scalare, il database sottostante.

[Per ulteriori informazioni di base sul servizio correlato Amazon Relational Database Service, consulta Amazon RDS](#)

ElastiCache può aiutarti a risparmiare sui costi del database archiviando i dati a cui accedi di frequente in una cache. Se l'applicazione ha requisiti di throughput di lettura elevati, è possibile ottenere scalabilità elevata, prestazioni veloci e costi di storage dei dati ridotti utilizzando ElastiCache, anziché scalare, il database sottostante.

Come ElastiCache funziona

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

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 offre due opzioni di implementazione: cluster serverless e cluster progettati autonomamente. Per informazioni, consulta [Scelta tra le opzioni di implementazione](#).

Note

Amazon ElastiCache funziona con i motori Valkey, Redis OSS e Memcached. Se non sei certo di quale motore utilizzare, consulta [Confronto tra le cache autoprogettate da ValkeyOSS, Redis e Memcached](#) in questa guida.

Argomenti

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

Come ElastiCache funziona

ElastiCache Senza server

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

Note

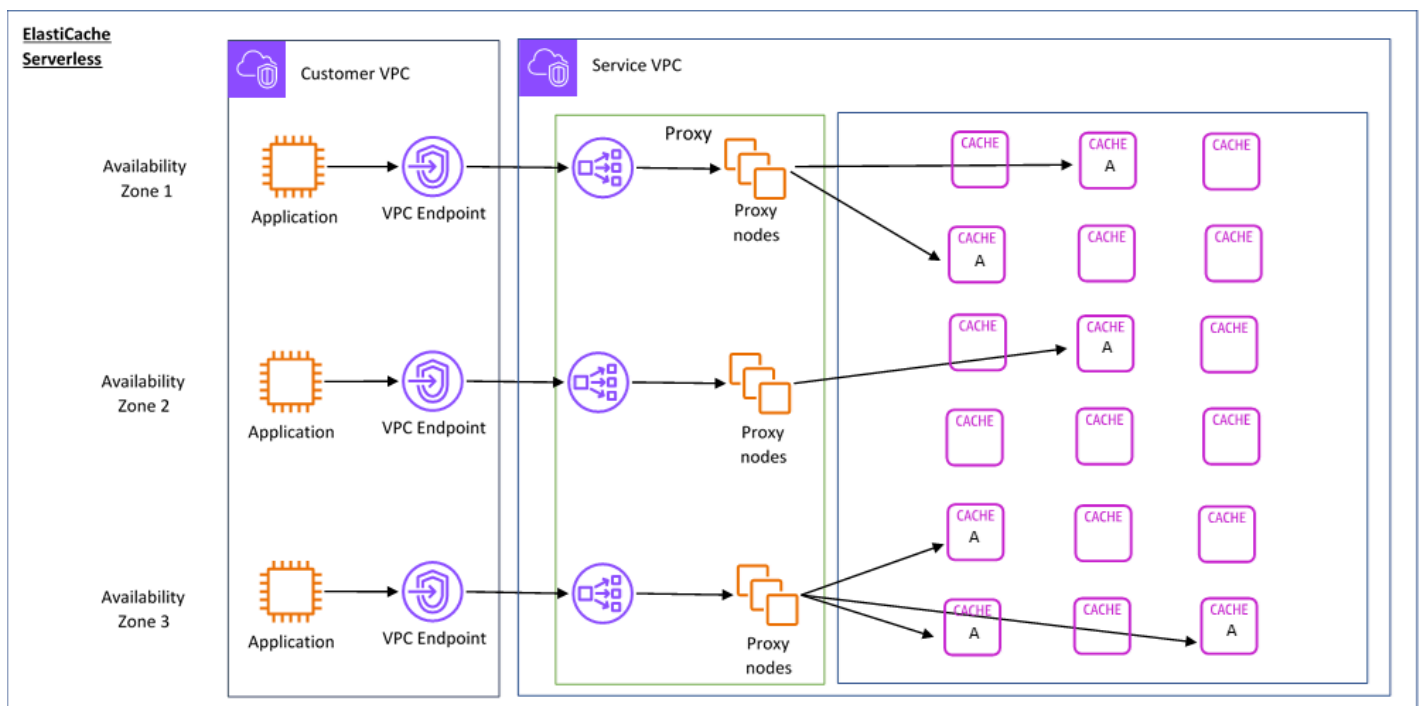
- ElastiCache Serverless esegue Valkey, Redis OSS o Memcached in modalità cluster ed è compatibile solo con i client che lo supportano. TLS

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 una disponibilità del 99,99% per ogni cache. SLA
- **Aggiornamenti automatici del software:** ElastiCache Serverless aggiorna automaticamente la cache all'ultima versione del software secondario e alle patch più recenti senza alcun impatto sulla disponibilità dell'applicazione. Quando sarà disponibile una nuova versione principale, ti ElastiCache 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 DNS endpoint 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.

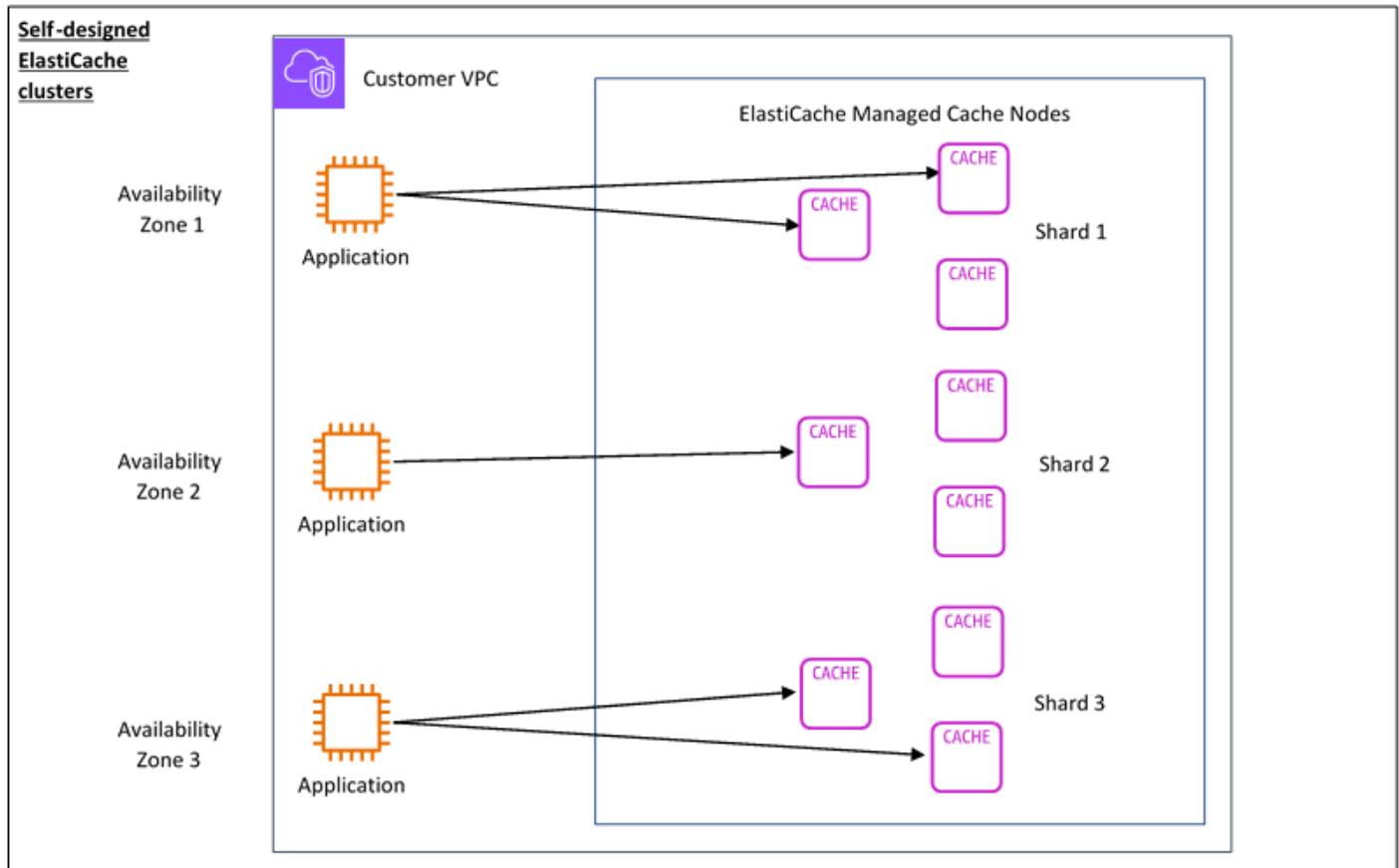
Cluster progettati autonomamente ElastiCache

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 Valkey o Redis OSS in modalità cluster creando un cluster con più shard o in modalità non cluster con un singolo shard.

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 distribuiti su più nodi AZs per ottenere un'elevata disponibilità.
- **Controllo granulare:** quando progetti il tuo cluster, hai un maggiore controllo sull'ottimizzazione delle impostazioni della cache. Ad esempio, è possibile utilizzare [Parametri Valkey e Redis OSS](#) o [Parametri Memcached specifici](#) 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 funzione Auto-Scaling per configurare il ridimensionamento in base a una pianificazione o il ridimensionamento in base a metriche come l'utilizzo della memoria nella CPU cache.

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



Prezzi delle dimensioni

È possibile eseguire la distribuzione ElastiCache in due opzioni di distribuzione. Quando si implementa ElastiCache Serverless, si paga in base all'utilizzo dei dati archiviati in GB all'ora e di calcolo in unità di elaborazione (ECPUs). ElastiCache ECPU Quando scegli di progettare i tuoi ElastiCache cluster, paghi per 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 (ECPUs)

Paghi per le richieste eseguite dalla tua applicazione su ElastiCache Serverless in ElastiCache Processing Units (ECPUs), un'unità che include sia v CPU time che i dati trasferiti.

- Le operazioni di lettura e scrittura semplici richiedono 1 ECPU per ogni kilobyte (KB) di dati trasferiti. Ad esempio, un GET comando che trasferisce fino a 1 KB di dati consuma 1. ECPU Una SET richiesta che trasferisce 3,2 KB di dati consumerà 3,2ECPUs.
- Con Valkey e RedisOSS, i comandi che consumano più CPU tempo di v e trasferiscono più dati consumano in ECPUs base alla maggiore delle due dimensioni. Ad esempio, se l'applicazione utilizza il HMGET comando, consuma 3 volte il CPU tempo v rispetto a un semplice GET comandoSET/e trasferisce 3,2 KB di dati, consumerà 3,2. ECPU In alternativa, se trasferisce solo 2 KB di dati, ne consumerà 3ECPUs.
- Con Valkey e RedisOSS, i comandi che richiedono un CPU tempo v aggiuntivo consumeranno proporzionalmente di più. ECPUs Ad esempio, se l'applicazione utilizza il OSS [HMGETcomando](#) Valkey o Redis e consuma 3 volte il CPU tempo v come semplice GET comandoSET/, consumerà 3 volte. ECPUs
- Con Memcached, i comandi che operano su più elementi consumeranno proporzionalmente di più. ECPUs Ad esempio, se l'applicazione esegue un multiget su 3 elementi, ne consumerà 3. ECPUs
- Con Memcached, i comandi che operano su più elementi e trasferiscono più dati consumano in ECPUs base alla più alta delle due dimensioni. Ad esempio, se l'applicazione utilizza il GET comando, recupera 3 elementi e trasferisce 3,2 KB di dati, consumerà 3,2. ECPU In alternativa, se trasferisce solo 2 KB di dati, ne consumerà 3ECPUs.

ElastiCache Serverless emette una nuova metrica chiamata `ElastiCacheProcessingUnits` che ti aiuta a comprendere il ECPUs consumo del tuo carico di lavoro.

Ore per nodo

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

ElastiCache backup

Un backup è una point-in-time copia di una cache serverless o di un cluster progettato autonomamente da Valkey o RedisOSS. 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 Valkey, Redis e Memcached supportati OSS 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 di disponibilità del 99,99% (). SLA I backup da OSS cluster Valkey o Redis progettati internamente possono essere ripristinati in una configurazione serverless.

Cluster progettati autonomamente

Se hai bisogno di un controllo granulare sul tuo cluster Valkey, Redis o Memcached, puoi scegliere di OSS progettare il tuo cluster 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 nelle 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 una disponibilità fino al 99,99%. SLA I backup dalle OSS cache Valkey o Redis senza server possono essere ripristinati in un cluster progettato autonomamente.

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 Valkey, Redis o MemcachedOSS, sul numero di nodi e sul posizionamento di tali 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
Versione supportata ElastiCache	Valkey 7.2 e versioni successive, Redis OSS versione 7.1 e successive, Memcached 1.6.21 e successive	Valkey 7.2 e versioni successive, Redis versione 4.0 e successive, Memcached 1.4 e successive OSS
Modalità cluster (Valkey e Redis) OSS	Fa funzionare i motori solo <code>incluster mode enabled</code> . I client devono supportare <code>cluster mode enabled</code> la connessione a ElastiCache Serverless.	Può essere configurato per funzionare in modalità cluster abilitata o disattivata in modalità cluster.

Funzionalità	Cache serverless	Cluster progettati autonomamente
Dimensionamento	<p>Ridimensiona automaticamente i motori sia verticalmente che orizzontalmente senza alcuna gestione della capacità.</p>	<p>Fornisce il controllo sulla scalabilità e richiede al contempo il monitoraggio per assicurarsi che la capacità attuale soddisfi adeguatamente la domanda.</p> <p>Per Valkey e RedisOSS, puoi scegliere di scalare verticalmente aumentando o diminuendo la dimensione del nodo di cache quando necessario. Puoi anche scalare orizzontalmente, aggiungendo nuovi shard o aggiungendo altre repliche ai tuoi shard. Questa funzionalità non è disponibile per Memcached.</p> <p>Con la funzione Auto-Scaling puoi anche configurare il ridimensionamento in base a una pianificazione o in base a metriche come CPU l'utilizzo della memoria nella cache.</p>
Connessione client	<p>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.</p>	<p>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.</p>

Funzionalità	Cache serverless	Cluster progettati autonomamente
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 motore .
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 il cluster in una singola zona di disponibilità o su più zone di disponibilità (). AZs Quando si utilizza Valkey o RedisOSS, fornisce ai cluster Multi-AZ dati 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 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 gestita da AWS o una chiave gestita dal cliente AWS KMS.
Crittografia in transito (TLS)	Sempre abilitata. I clienti devono supportare la TLS connettività.	Opzione per abilitare o disabilitare.

Funzionalità	Cache serverless	Cluster progettati autonomamente
Backup	<p>Supporta backup automatici e manuali delle cache senza alcun impatto sulle prestazioni.</p> <p>I OSS backup Valkey e Redis sono compatibili tra loro e possono essere ripristinati in una cache ElastiCache Serverless o in un cluster progettato autonomamente.</p>	<p>Supporta backup automatici e manuali per Valkey e Redis. OSS I cluster possono subire un certo impatto sulle prestazioni a seconda della memoria riservata disponibile. Per ulteriori informazioni, consulta Gestione della memoria riservata per Valkey e Redis OSS.</p> <p>I OSS backup Valkey e Redis sono compatibili tra loro e possono essere ripristinati in una cache ElastiCache Serverless o in un cluster progettato autonomamente.</p>

Funzionalità	Cache serverless	Cluster progettati autonomamente
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 il ECPUs consumo.</p> <p>ElastiCache Serverless invia eventi indicando 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 per eventi significativi. SNS Consulta Parametri per Memcached e Metriche per Valkey e Redis OSS.</p>
Disponibilità	Contratto sul livello di servizio di disponibilità del 99,99% () SLA	I cluster progettati autonomamente possono essere progettati per raggiungere il 99,99% di disponibilità del Service Level Agreement (SLA) , a seconda della configurazione.

Funzionalità	Cache serverless	Cluster progettati autonomamente
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 correzioni 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 memoria e storage locale SSD (unità a stato solido). Il data tiering offre un'opzione in termini di rapporto prezzo/prestazioni per i OSS carichi di lavoro Valkey e Redis utilizzando unità a stato solido (SSDs) 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 (E). ElastiCache ECPU 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 ElastiCache cluster](#)

ElastiCache Risorse Amazon per utenti alle prime armi

Consigliamo agli utenti alle prime armi di iniziare a leggere le seguenti sezioni e di farvi riferimento se necessario.

- Aspetti salienti del servizio e prezzi: la [pagina dei dettagli del prodotto](#) fornisce una panoramica generale del prodotto ElastiCache, dei punti salienti del servizio e dei prezzi.

- ElastiCache video: la [ElastiCache Video](#) sezione contiene video che ti introducono ad Amazon ElastiCache. I video illustrano casi d'uso comuni ElastiCache e dimostrano come utilizzarli per ElastiCache ridurre la latenza e migliorare la velocità effettiva delle applicazioni.
- Nozione di base – La [Guida introduttiva ad Amazon ElastiCache](#) 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 larga scala: il ElastiCache white paper [Performance at scale with Amazon](#) affronta le strategie di caching che aiutano la tua applicazione a funzionare bene su larga scala.

Al termine delle sezioni precedenti, leggi queste sezioni:

- [Scelta delle dimensioni dei nodi](#)

Vuoi che i nodi siano abbastanza grandi da ospitare tutti i dati da memorizzare nella cache. Allo stesso tempo, non vuoi pagare per cache non necessaria. Puoi utilizzare questo argomento per scegliere la dimensione ideale dei nodi.

- [ElastiCache migliori pratiche e strategie di caching](#)

Identificare e risolvere problemi che possono influire sull'efficienza del cluster.

Se desideri utilizzare il AWS Command Line Interface (AWS CLI), puoi utilizzare questi documenti per aiutarti a iniziare:

- [AWS Command Line Interface documentazione](#)

Questa sezione fornisce informazioni su come scaricare AWS CLI, far AWS CLI funzionare il sistema e fornire le AWS credenziali.

- [AWS CLI documentazione per ElastiCache](#)

Questo documento separato copre tutti i ElastiCache comandi AWS CLI for, inclusi la sintassi e gli esempi.

È possibile scrivere programmi applicativi da utilizzare ElastiCache API con una varietà di linguaggi di programmazione popolari. Ecco alcune risorse:

- [Strumenti per Amazon Web Services](#)

Amazon Web Services fornisce una serie di kit di sviluppo software (SDKs) con supporto per ElastiCache. Puoi programmare per ElastiCache usare Java, .NET, PHP, Ruby e altri linguaggi. Questi SDKs possono semplificare notevolmente lo sviluppo delle applicazioni formattando le richieste ElastiCache, analizzando le risposte e fornendo la logica dei tentativi e la gestione degli errori.

- [Usando il ElastiCache API](#)

Se non desideri utilizzare il AWS SDKs, puoi interagire ElastiCache direttamente con Query. API In questa sezione puoi trovare suggerimenti per risolvere i problemi e informazioni su come creare e autenticare richieste e gestire risposte.

- [ElastiCache API Riferimento Amazon](#)

Questo documento separato copre tutte le ElastiCache API operazioni, inclusi la sintassi e gli esempi.

AWS Regioni 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. Ogni sede del data center è denominata AWS Regione.

Ogni AWS regione contiene più località distinte chiamate zone di disponibilità o AZs. Ogni zona di disponibilità è progettata per rimanere isolata dai guasti che si verificano in altre zone di disponibilità. Ciascuna è progettata per fornire connettività di rete economica 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, consulta [Ridurre al minimo i tempi di inattività in ElastiCache \(Redis\) con Multi-AZ](#). OSS

Casi ElastiCache d'uso comuni e come ElastiCache può essere utile

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 la query o la API chiamata remota al database più ottimizzata sono notevolmente più lente rispetto al recupero di una chiave piatta da una cache in memoria. I tempi notevolmente più lenti spingono i clienti ad andare altrove.

Gli esempi seguenti illustrano alcuni dei modi in cui l'utilizzo ElastiCache può migliorare le prestazioni complessive dell'applicazione.

Argomenti

- [Datastore in memoria](#)
- [Classifiche di gioco](#)
- [Messaggistica \(Pub/Sub\)](#)
- [Dati di raccomandazione \(hash\)](#)
- [ElastiCache Testimonianze dei clienti](#)

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.

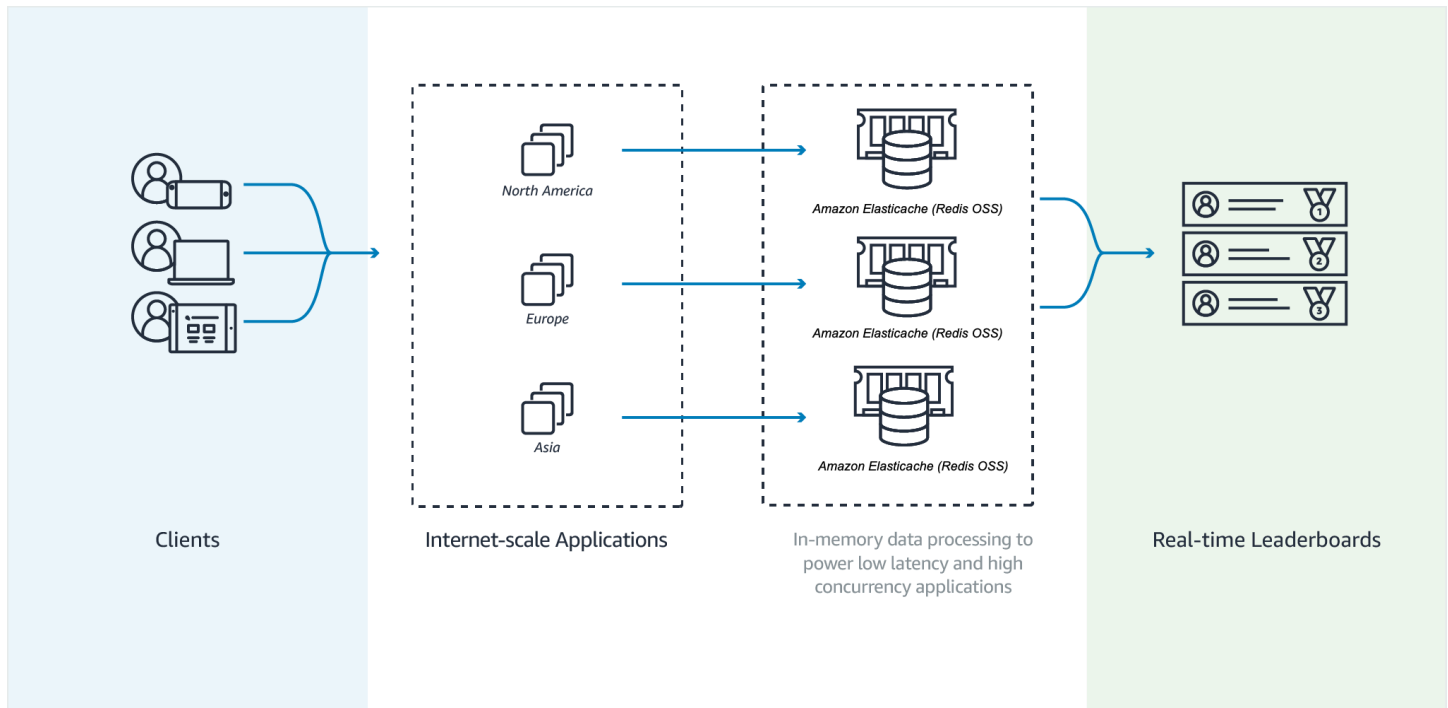
Per ulteriori informazioni, consulta [Strategie di caching per Memcached](#)

Classifiche di gioco

Con i set OSS ordinati Valkey o Redis puoi spostare la complessità computazionale delle classifiche dall'applicazione al cluster.

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 Valkey e Redis garantiscono sia l'unicità che l'ordine degli elementi. Con i set ordinati, ogni volta che un nuovo elemento viene aggiunto al set ordinato, questo viene riclassificato in tempo reale. Viene quindi aggiunto al set nel corretto ordine numerico.

Nel diagramma seguente, puoi vedere come funziona una classifica di gioco. ElastiCache



Example Classificazione Valkey o Redis OSS

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 a base zero, ZREVRANK restituisce 1 per June, che è in seconda posizione.

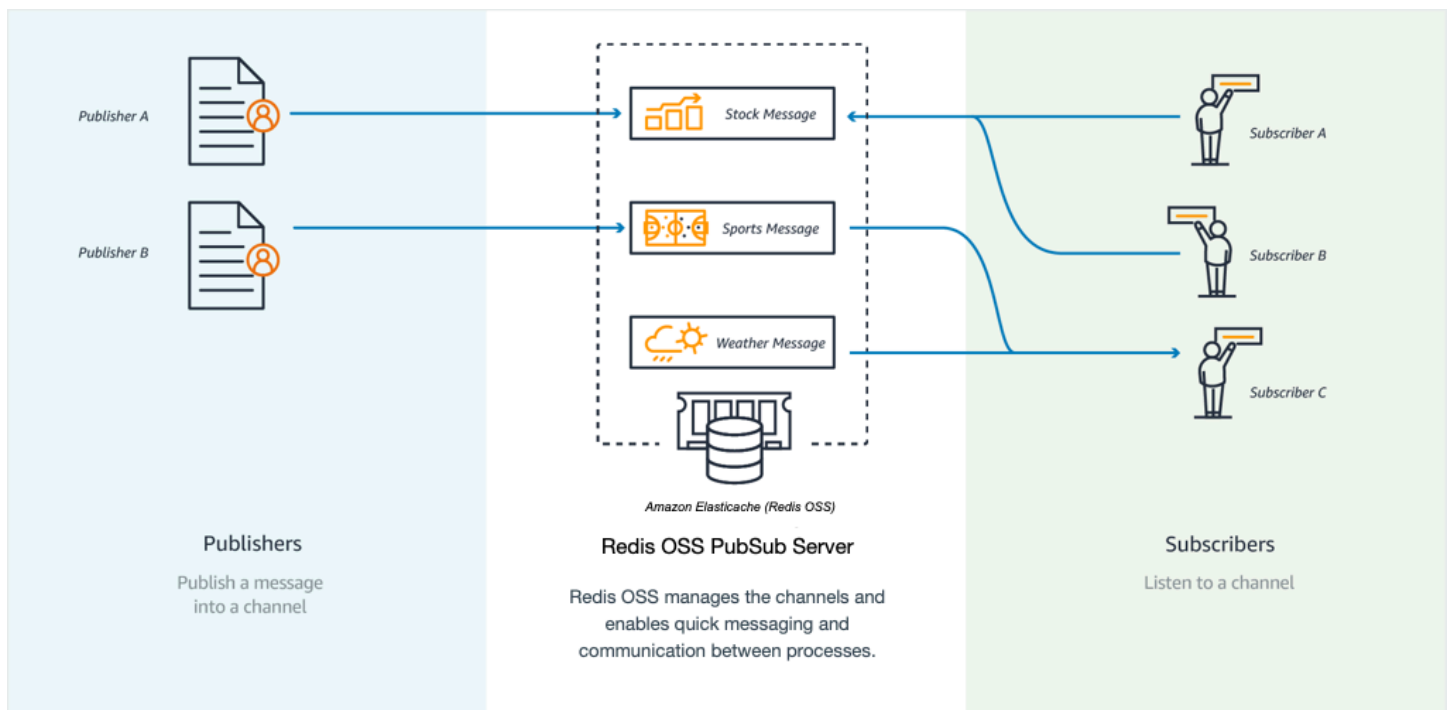
```
ZREVRANK leaderboard June
1
```

Per ulteriori informazioni, consulta la [documentazione di Valkey](#) sugli insiemi ordinati.

Messaggistica (Pub/Sub)

Quando invii un messaggio e-mail, lo invii a uno o più destinatari specificati. Nel paradigma OSS pub/sub di Valkey e Redis, si invia un messaggio a un canale specifico senza sapere chi, se qualcuno, 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 funzionalità Pub/sub non ha alcuna relazione con lo spazio chiave. Perciò, non interferisce a nessun livello. Nel diagramma seguente, puoi trovare un'illustrazione della ElastiCache messaggistica con Valkey e Redis. OSS



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 iscriverti a un singolo canale, usa il SUBSCRIBE comando che specifica il canale a cui vuoi iscriverti. Nell'esempio seguente, un cliente effettua la sottoscrizione al canale news.sports.golf.

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

Dopo un po', il client annulla l'iscrizione al canale utilizzando il UNSUBSCRIBE comando che specifica il canale da cui annullare l'iscrizione.

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

Example – Sottoscrizioni a più canali specificati

Per iscriverti a più canali specifici, 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 un abbonamento a un canale specifico, usa il UNSUBSCRIBE comando e specifica il canale da cui annullare l'iscrizione.

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

Per annullare gli abbonamenti a più canali, usa il UNSUBSCRIBE comando e specifica i canali da cui annullare l'iscrizione.

```
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 abbonarsi a tutti i canali che corrispondono a uno schema utilizzando il PSUBSCRIBE comando.

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

La stringa del canale inviata a un SUBSCRIBE comando [P] e al UNSUBSCRIBE comando [P] deve 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 client non può pubblicare su un canale a cui è abbonato.

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

Dati di raccomandazione (hash)

L'utilizzo di INCR o DECR in Valkey o Redis OSS semplifica la compilazione dei consigli. 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`. Utilizzando gli hash, puoi anche mantenere un elenco di tutti coloro che hanno apprezzato o non apprezzato 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
```

ElastiCache Testimonianze dei clienti

Per scoprire in che modo aziende come AirbnbPBS, Esri e altre utilizzano Amazon ElastiCache per far crescere le proprie attività con una migliore esperienza cliente, consulta [Come gli altri utilizzano Amazon ElastiCache](#).

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

Guida introduttiva ad Amazon ElastiCache

Usa il tutorial pratico in questa sezione per iniziare e saperne di più sull'utilizzo. ElastiCache

Argomenti

- [Configurazione ElastiCache](#)
- [Crea una cache serverless Valkey](#)
- [Crea una cache serverless Valkey o Redis OSS](#)
- [Crea una cache serverless Memcached](#)
- [Tutorial: Guida introduttiva a Python e ElastiCache](#)
- [Tutorial: Configurazione di Lambda per ElastiCache l'accesso in un VPC](#)

Configurazione ElastiCache

Per utilizzare il servizio ElastiCache web, segui questi passaggi.

Argomenti

- [Registrati per un Account AWS](#)
- [Crea un utente con accesso amministrativo](#)
- [Concessione dell'accesso programmatico](#)
- [Configura le tue autorizzazioni \(solo nuovi ElastiCache utenti\)](#)
- [Configurare EC2](#)
- [Concedi l'accesso di rete da un gruppo VPC di sicurezza Amazon alla tua cache](#)
- [Scarica e configura l'accesso alla riga di comando](#)

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 <https://portal.aws.amazon.com/billing/registrazione>.
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 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. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a <https://aws.amazon.com/> e scegliendo Il mio account.

Crea un utente con accesso amministrativo

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

Proteggi i tuoi Utente root dell'account AWS

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

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

2. Attiva l'autenticazione a più fattori (MFA) per il tuo utente root.

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

Crea un utente con accesso amministrativo

1. Abilita IAM Identity Center.

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

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

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

Accesso come utente amministratore

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

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

Assegna l'accesso a ulteriori utenti

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

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

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

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

Concessione dell'accesso programmatico

Gli utenti necessitano di un accesso programmatico se desiderano interagire con AWS l'esterno di AWS Management Console 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	Utilizza credenziali temporane e per firmare le richieste	Segui le istruzioni per l'interfaccia che desideri utilizzare.

Quale utente necessita dell'accesso programmatico?	Per	Come
(Utenti gestiti in IAM Identity Center)	programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	<ul style="list-style-type: none">• Per la AWS CLI, vedere Configurazione dell'uso AWS IAM Identity Center nella AWS CLI Guida per l'utente.AWS Command Line Interface• Per AWS SDKs gli strumenti e AWS APIs, consulta l'autenticazione di IAM Identity Center nella Guida di riferimento agli strumenti AWS SDKs e agli strumenti.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	Seguendo le istruzioni riportate in Utilizzo delle credenziali temporanee con le AWS risorse nella Guida per l'IAM utente.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare richieste programmatiche a AWS CLI,, AWS SDKs o. AWS APIs	<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 IAM utente nella Guida per l'utente.AWS Command Line Interface • Per AWS SDKs gli strumenti , consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli strumenti e agli AWS SDKs strumenti. • Per AWS APIs, consulta Gestione delle chiavi di accesso per IAM gli utenti nella Guida per l'IAMutente.

Argomenti correlati:

- [Cosa contiene IAM la](#) Guida IAM per l'utente.
- AWS Informazioni AWS generali [sulle credenziali di sicurezza](#).

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 IAM tramite un provider di identità:

Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate in [Creazione di un ruolo per un provider di identità di terze parti \(federazione\)](#) nella Guida per l'IAMutente.

- IAMutenti:
 - Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella sezione [Creazione di un ruolo per un IAM utente](#) nella Guida per l'IAMutente.
 - (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate in [Aggiungere autorizzazioni a un utente \(console\)](#) nella Guida per l'IAMutente.

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, assicurati di disporre delle autorizzazioni per la chiamata `iam:createServiceLinkedRole` o di aver creato il ruolo collegato al servizio. ElastiCache

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Creazione di una nuova politica \(\) IAM](#)
- [AWS politiche gestite per Amazon ElastiCache](#)
- [Utilizzo di ruoli collegati ai servizi per Amazon ElastiCache](#)

Configurare EC2

Dovrai configurare un'EC2istanza da cui connetterti alla cache.

- Se non disponi già di un'EC2istanza, scopri come configurarla EC2 qui: [Guida introduttiva EC2](#).
- L'EC2istanza deve trovarsi nella stessa istanza VPC e avere le stesse impostazioni del gruppo di sicurezza della cache. Per impostazione predefinita, Amazon ElastiCache crea una cache predefinita VPC e utilizza il gruppo di sicurezza predefinito. Per seguire questo tutorial, assicurati che la tua EC2 istanza sia nell'impostazione predefinita VPC e abbia il gruppo di sicurezza predefinito.

Concedi l'accesso di rete da un gruppo VPC di sicurezza Amazon alla tua cache

ElastiCache i cluster progettati autonomamente utilizzano la porta 6379 per i OSS comandi Valkey e Redis, mentre i ElastiCache server serverless utilizzano sia la porta 6379 che la porta 6380. Per connettere ed eseguire correttamente i OSS comandi Valkey o Redis dall'EC2istanza, il gruppo di sicurezza deve consentire l'accesso a queste porte secondo necessità.

ElastiCache (Memcached) utilizza le porte 11211 e 11212 per accettare i comandi Memcached. Per connettere ed eseguire correttamente i comandi Memcached dall'EC2istanza, il gruppo di sicurezza deve consentire l'accesso a queste porte.

1. Accedi AWS Command Line Interface e apri la [EC2console Amazon](#).
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 tuo AmazonVPC. 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 TCPRegola personalizzata.
- d. Se usi Valkey o RedisOSS, nella casella Intervallo di porte, digita. 6379

Se si utilizza Memcached, nella casella Intervallo di porte, digitare. 11211

- e. Nella casella Source, scegli Anywhere con l'intervallo di porte (0.0.0.0/0) in modo che qualsiasi EC2 istanza Amazon che avvii all'interno di Amazon VPC possa connettersi alla cache.
- f. Se utilizzi ElastiCache serverless, aggiungi un'altra regola scegliendo Aggiungi regola.
- g. Nella colonna Tipo, scegli TCP Regola personalizzata.
- h. Se usi ElastiCache (RedisOSS), nella casella Intervallo di porte, digita6380.

Se si utilizza ElastiCache (Memcached), nella casella Intervallo di porte, digitare. 11212

- i. Nella casella Source, scegli Anywhere con l'intervallo di porte (0.0.0.0/0) in modo che qualsiasi EC2 istanza Amazon che avvii all'interno di Amazon VPC possa connettersi alla cache.

j. Seleziona Salva

Scarica e configura l'accesso alla riga di comando

Scarica e installa l'utilità valkey-cli.

Se la usi ElastiCache con Valkey, potresti trovare utile l'utilità valkey-cli. Se utilizzi ElastiCache (RedisOSS) con redis-cli, valuta la possibilità di passare a valkey-cli poiché funziona anche per Redis. OSS

1. Connettiti alla tua EC2 istanza Amazon utilizzando l'utilità di connessione che preferisci. Per istruzioni su come connettersi a un'EC2 istanza Amazon, consulta la [Amazon EC2 Getting Started Guide](#).
2. Scarica e installa l'utilità valkey-cli eseguendo il comando appropriato per la tua 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
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
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 alla build per l'installazione di valkey-cli o TLS redis-cli. ElastiCache Serverless è accessibile solo quando è abilitato. TLS
- Se ci si connette a un cluster non crittografato, non è necessaria l'opzione `Build_TLS=yes`.

Crea una cache serverless Valkey

In questo passaggio, crei una nuova cache in Amazon ElastiCache.

AWS Management Console

Per creare una nuova cache utilizzando la ElastiCache console:

1. Accedi a AWS Management Console e apri il <https://console.aws.amazon.com/connect/>.
2. Nel riquadro di navigazione sul lato sinistro della console, scegli Valkey caches.
3. Sul lato destro della console, scegli Crea cache Valkey
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 la cache è nello stato ACTIVE "", puoi iniziare a scrivere e leggere i dati nella cache.

AWS CLI

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

Linux

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

Windows

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

Il valore del campo Stato è impostato su CREATING.

Per verificare di aver ElastiCache terminato la creazione della cache, utilizzare il describe-serverless-caches comando.

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 [Leggi e scrivi dati nella cache](#).

Leggi e scrivi dati nella cache

Questa sezione presuppone che tu abbia creato un'EC2istanza Amazon e che tu possa connetterti ad essa. Per istruzioni su come eseguire questa operazione, consulta la [Amazon EC2 Getting Started Guide](#).

Questa sezione presuppone inoltre che tu abbia configurato VPC l'accesso e le impostazioni del gruppo di sicurezza per l'EC2istanza da cui ti connetti alla cache e che tu abbia configurato valkey-cli sull'istanza. EC2 Per ulteriori informazioni su questa fase, consulta [Configurazione ElastiCache](#).

Ricerca dell'endpoint della cache

AWS Management Console

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

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione sul lato sinistro della console, scegli Valkey caches.
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' AWS CLI esempio 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
```

Connect a Valkey Cache (Linux)

Ora che hai l'endpoint di cui hai bisogno, puoi accedere alla tua EC2 istanza e connetterti alla cache. Nell'esempio seguente, si utilizza l'utilità `valkey-cli` per connettersi a un cluster. Il comando seguente esegue la connessione a una cache (nota: sostituisci `cache-endpoint` con l'endpoint recuperato nella fase precedente).

```
src/valkey-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"
```

Connect a Valkey Cache (Windows)

Ora che hai l'endpoint di cui hai bisogno, puoi accedere alla tua EC2 istanza e connetterti alla cache. Nell'esempio seguente, si utilizza l'utilità `valkey-cli` per connettersi a un cluster. Il seguente comando esegue la connessione a una cache. Apri il prompt dei comandi, passa alla OSS directory Valkey o Redis ed esegui il comando (nota: sostituisci `Cache_Endpoint` con l'endpoint recuperato nel passaggio precedente).

```
c:\Valkey>valkey-cli -h Valkey_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 [\(Facoltativo\) Pulizia](#).

(Facoltativo) Pulizia

Se non ti serve più la ElastiCache cache Amazon che hai creato, puoi eliminarla. In questo modo hai la certezza che non ti vengano addebitati costi per risorse che non stai utilizzando. Puoi usare la ElastiCache console AWS CLI, o il ElastiCache API per eliminare la cache.

AWS Management Console

Per eliminare la cache tramite la console:

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione sul lato sinistro della console, scegli Valkey Caches.
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 DELETINGallo stato attuale, smetti di incorrere in addebiti.

AWS CLI

L' AWS CLI esempio 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
```

Si noti che il valore del campo Status è impostato DELETINGsu.

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

Fasi successive

Per maggiori informazioni su questo argomento, ElastiCache consulta le seguenti pagine:

- [Lavorare con ElastiCache](#)
- [Ridimensionamento ElastiCache](#)
- [Registrazione e monitoraggio in Amazon ElastiCache](#)
- [ElastiCache migliori pratiche e strategie di caching](#)
- [Snapshot e ripristino](#)
- [SNSMonitoraggio degli ElastiCache eventi da Amazon](#)

Crea una cache serverless Valkey o Redis OSS

In questo passaggio, crei una nuova cache in Amazon ElastiCache.

AWS Management Console

Per creare una nuova cache utilizzando la ElastiCache console:

1. Accedi a AWS Management Console e apri il <https://console.aws.amazon.com/connect/>.
2. Nel riquadro di navigazione sul lato sinistro della console, scegli Cache Valkey o cache OSS Redis.
3. Sul lato destro della console, scegli Crea cache Valkey o Crea cache Redis OSS
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 la cache è nello stato ACTIVE "", puoi iniziare a scrivere e leggere i dati nella cache.

AWS CLI

L' AWS CLI esempio 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 di aver ElastiCache terminato la creazione della cache, utilizzare il `describe-serverless-caches` comando.

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 [Leggi e scrivi dati nella cache](#).

Leggi e scrivi dati nella cache

Questa sezione presuppone che tu abbia creato un'EC2istanza Amazon e che tu possa connetterti ad essa. Per istruzioni su come eseguire questa operazione, consulta la [Amazon EC2 Getting Started Guide](#).

Questa sezione presuppone inoltre che tu abbia configurato VPC l'accesso e le impostazioni del gruppo di sicurezza per l'EC2istanza da cui ti connetti alla cache e che tu abbia configurato valkey-cli sull'istanza. EC2 Per ulteriori informazioni su questa fase, consulta [Configurazione ElastiCache](#).

Ricerca dell'endpoint della cache

AWS Management Console

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

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione sul lato sinistro della console, scegli Valkey caches OSS 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' AWS CLI esempio 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
```

Connect a Valkey o Redis OSS Cache (Linux)

Ora che hai l'endpoint di cui hai bisogno, puoi accedere alla tua EC2 istanza e connetterti alla cache. Nell'esempio seguente, si utilizza l'utilità `valkey-cli` per connettersi a un cluster. Il comando seguente esegue la connessione a una cache (nota: sostituisci `cache-endpoint` con l'endpoint recuperato nella fase precedente).

```
src/valkey-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"
```

Connect a Valkey o Redis OSS Cache (Windows)

Ora che hai l'endpoint di cui hai bisogno, puoi accedere alla tua EC2 istanza e connetterti alla cache. Nell'esempio seguente, si utilizza l'utilità `valkey-cli` per connettersi a un cluster. Il seguente comando esegue la connessione a una cache. Aprite il prompt dei comandi, passate alla directory Valkey

ed eseguite il comando (nota: sostituite `Cache_Endpoint` con l'endpoint recuperato nel passaggio precedente).

```
c:\Redis>valkey-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 [\(Facoltativo\) Pulizia](#).

(Facoltativo) Pulizia

Se non ti serve più la ElastiCache cache Amazon che hai creato, puoi eliminarla. In questo modo hai la certezza che non ti vengano addebitati costi per risorse che non stai utilizzando. Puoi utilizzare la ElastiCache console, il AWS CLI, o il ElastiCache API per eliminare la cache.

AWS Management Console

Per eliminare la cache tramite la console:

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione sul lato sinistro della console, scegli Valkey o Redis OSS Caches.
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 DELETINGallo stato attuale, smetti di incorrere in addebiti.

AWS CLI

L' AWS CLI esempio 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
```

Si noti che il valore del campo Status è impostato DELETINGsu.

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

Fasi successive

Per maggiori informazioni su questo argomento, ElastiCache consulta le seguenti pagine:

- [Lavorare con ElastiCache](#)
- [Ridimensionamento ElastiCache](#)
- [Registrazione e monitoraggio in Amazon ElastiCache](#)
- [ElastiCache migliori pratiche e strategie di caching](#)
- [Snapshot e ripristino](#)
- [SNSMonitoraggio degli ElastiCache eventi da Amazon](#)

Crea una cache serverless Memcached

AWS Management Console

Per creare una nuova cache serverless Memcached utilizzando la console: ElastiCache

1. Accedi AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione sul lato sinistro della console scegli Cache Memcached.
3. Sul lato destro della console scegli Crea cache Memcached.
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.

- Una volta che lo stato della cache è ACTIVE "", puoi iniziare a scrivere e leggere i dati nella cache.

Per creare una nuova cache utilizzando AWS CLI

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

Linux

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

Windows

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

Il valore del campo Stato è impostato su CREATING.

Per verificare di aver ElastiCache terminato la creazione della cache, utilizzare il describe-serverless-caches comando.

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 [Leggi e scrivi dati nella cache](#).

Leggi e scrivi dati nella cache

Questa sezione presuppone che tu abbia creato un'EC2istanza Amazon e che tu possa connetterti ad essa. Per istruzioni su come eseguire questa operazione, consulta la [Amazon EC2 Getting Started Guide](#).

Per impostazione predefinita, ElastiCache crea una cache come impostazione predefinitaVPC. Assicurati che anche l'EC2istanza sia creata nell'impostazione predefinitaVPC, in modo che sia in grado di connettersi alla cache.

Ricerca dell'endpoint della cache

AWS Management Console

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

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione sul lato sinistro della console scegli Cache Memcached.
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' AWS CLI esempio seguente mostra come trovare l'endpoint per la nuova cache utilizzando il `describe-serverless-caches` comando. 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
```

Connect usando Open SSL

Per informazioni su come connettersi tramite OpenSSL, vedere [ElastiCache crittografia in transito \(\) TLS](#)

Connessione tramite il client Memcached Java

Per informazioni su come eseguire la connessione tramite il client Memcached Java, consulta [ElastiCache crittografia in transito \(\) TLS](#)

Connect utilizzando il client Memcached PHP

```
<?php
$cluster_endpoint = "mycluster.serverless.use1.cache.amazonaws.com";
$server_port = 11211;

/* Initialize a persistent Memcached client in TLS mode */
$tls_client = new Memcached('persistent-id');
$tls_client->addServer($cluster_endpoint, $server_port);
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.serverless.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;
$tls_client->createAndSetTLSContext((array)$tls_config);

/* store the data for 60 seconds in the cluster */
$tls_client->set('key', 'value', 60);
?>
```

Connessione tramite il client Memcached Python (Pymemcache)

[Vedi `memcached.readthedocs.io/en/latest/getting_started.html`](https://memcached.readthedocs.io/en/latest/getting_started.html) [https://py](https://pymemcache.org/)

```
import ssl
from pymemcache.client.base import Client

context = ssl.create_default_context()
cluster_endpoint = <To be taken from the AWS CLI / console>
target_port = 11211
memcached_client = Client("{cluster_endpoint}", target_port, tls_context=context)
memcached_client.set("key", "value", expire=500, noreply=False)
assert self.memcached_client.get("key").decode() == "value"
```

Connessione tramite il client Memcached NodeJS/TS (Electrode-IO memcache)

[Vedi `https://github.com/electrode-io/memcache`](https://github.com/electrode-io/memcache) e [memcache-client `https://www.npmjs.com/package/`](https://www.npmjs.com/package/memcache-client)

Installa usando `npm i memcache-client`.

Nell'applicazione, crea un client memcached come segue: TLS

```
var memcache = require("memcache-client");
const client = new memcache.MemcacheClient({server: "{cluster_endpoint}:11211", tls:
  {}});
client.set("key", "value");
```

Connessione tramite il client Memcached Rust (rust-memcache)

Vedi <https://crates.io/crates/memcache> e <https://github.com/aisk/rust-memcache>.

```
// create connection with to memcached server node:
let client = memcache::connect("memcache+tls://<cluster_endpoint>:11211?
verify_mode=none").unwrap();

// set a string value
client.set("foo", "bar", 0).unwrap();
```

Connessione tramite il client Memcached Go (Gomemcache)

<https://github.com/bradfitz/Vedi> gomemcache

```
c := New(net.JoinHostPort("{cluster_endpoint}", strconv.Itoa(port)))
c.DialContext = func(ctx context.Context, network, addr string) (net.Conn, error) {
var td tls.Dialer
td.Config = &tls.Config{}
return td.DialContext(ctx, network, addr)
}
foo := &Item{Key: "foo", Value: []byte("fooval"), Flags: 123}
err := c.Set(foo)
```

Connessione tramite il client Memcached Ruby (Dalli)

Vedi dalli <https://github.com/petergoldstein/>

```
require 'dalli'
ssl_context = OpenSSL::SSL::SSLContext.new
ssl_context.ssl_version = :SSLv23
ssl_context.verify_hostname = true
ssl_context.verify_mode = OpenSSL::SSL::VERIFY_PEER
client = Dalli::Client.new("<cluster_endpoint>:11211", :ssl_context => ssl_context);
```



```
client.get("abc")
```

Connect tramite Memcached. NETcliente () EnyimMemcachedCore

Vedi <https://github.com/cnblogs/EnyimMemcachedCore>

```
"MemcachedClient": {  
  "Servers": [  
    {  
      "Address": "{cluster_endpoint}",  
      "Port": 11211  
    }  
  ],  
  "UseSslStream": true  
}
```

Ora puoi procedere alla [\(Facoltativo\) Pulizia](#).

(Facoltativo) Pulizia

Usando il AWS Management Console

La procedura seguente elimina una sola cache dall'implementazione. Per eliminare più cache, ripeti la procedura per ogni cache da eliminare. Non occorre attendere la fine dell'eliminazione di una cache prima di avviare la procedura per eliminarne un'altra.

Per eliminare una cache

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nella dashboard della ElastiCache console, scegli il motore su cui è in esecuzione la cache che desideri eliminare. Viene visualizzato un elenco di tutte le cache che eseguono quel motore.
3. Per scegliere la cache da eliminare, seleziona il nome della cache nell'elenco.

Important

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

4. In Actions (Azioni), scegliere Delete (Elimina).
5. Nella schermata di conferma Elimina cache scegli Elimina per eliminare la cache o scegli Annulla per mantenerla.
6. Se si sceglie Elimina, lo stato della cache diventa Eliminazione in corso.

Non appena la cache passa DELETINGallo stato attuale, smetti di incorrere in addebiti per essa.

Usando il AWS CLI

Il seguente codice elimina la cache my-cache.

```
aws elasticache delete-serverless-cache --serverless-cache-name my-cache
```

L' `delete-serverless-cache`CLIazione elimina solo una cache serverless. Per eliminare più cache, richiama `delete-serverless-cache` ogni cache serverless che desideri eliminare. Non occorre attendere la fine dell'eliminazione di una cache serverless prima di eliminarne un'altra.

Per Linux, macOS o Unix:

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

Per Windows:

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

Per ulteriori informazioni, consulta l'argomento AWS CLI per ElastiCache . `delete-serverless-cache`

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

Fasi successive

Per ulteriori informazioni su ElastiCache vedi:

- [Lavorare con ElastiCache](#)
- [Ridimensionamento ElastiCache](#)
- [Quote per ElastiCache](#)
- [ElastiCache migliori pratiche e strategie di caching](#)
- [Visualizzazione ElastiCache degli eventi](#)

Tutorial: Guida introduttiva a Python e ElastiCache

Questa sezione contiene tutorial pratici per aiutarti a imparare a usare Valkey e Redis. ElastiCache OSS Ti raccomandiamo di lavorare con dei tutorial specifici della lingua.

Note

AWS SDKs sono disponibili per un'ampia varietà di lingue. Per un elenco completo consulta la pagina relativa agli [Strumenti per Amazon Web Services](#).

Argomenti

- [Python e ElastiCache](#)

Python e ElastiCache

In questo tutorial, usi AWS SDK for Python (Boto3) per scrivere semplici programmi per eseguire le seguenti operazioni ElastiCache (Redis): OSS

- Crea cluster ElastiCache (RedisOSS) (modalità cluster abilitata e modalità cluster disabilitata)
- Controlla se esistono utenti o gruppi di utenti, altrimenti creali. (Questa funzionalità è disponibile con Valkey 7.2 e versioni successive e con OSS Redis 6.0 e versioni successive.)
- Connect a 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.

Mentre segui questo tutorial, puoi fare riferimento alla documentazione AWS SDK di Python (Boto).

[La sezione seguente è specifica per ElastiCache: client di basso livello ElastiCache](#)

Prerequisiti dei tutorial

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

Argomenti

- [Tutorial: creazione di cluster e utenti ElastiCache](#)
- [Tutorial: Connessione a ElastiCache](#)
- [Esempi di utilizzo](#)

Tutorial: creazione di cluster e utenti ElastiCache

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

Argomenti

- [Creare un cluster in modalità cluster disabilitata](#)
- [Crea un cluster disattivato in modalità cluster con e TLS RBAC](#)

- [Creare un cluster in modalità cluster abilitata](#)
- [Crea un cluster abilitato alla modalità cluster con e TLS 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/dg/
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='valkey',
        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, Valkey 7.2, one primary and two replicas
    elasticacheResponse = create_cluster_mode_disabled(
        #CacheNodeType='cache.m6g.large',
        EngineVersion='7.2',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104053'
    )

    logging.info(elasticacheResponse)
```

Per eseguire il programma, immetti il comando seguente:

```
python CreateClusterModeDisabledCluster.py
```

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

Crea un cluster disattivato in modalità cluster con e TLS RBAC

Per garantire la sicurezza, è possibile utilizzare Transport Layer Security (TLS) e Role-Based Access Control (RBAC) durante la creazione di un cluster disattivato in modalità cluster. A differenza di Valkey o Redis OSSAUTH, in cui tutti i client autenticati hanno 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 degli accessi basato sui ruoli \(\) RBAC](#).

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

```
import boto3
import logging
```

```

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

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/dg/
  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': 'valkey',
          '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, Valkey 7.2, 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='7.2',
        NumCacheClusters=3,
        ReplicationGroupDescription='Valkey cluster mode disabled with replicas',
        ReplicationGroupId='valkey202104',
        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 in ElastiCache](#).

Creare un cluster in modalità cluster abilitata

Copiate il seguente programma e incollatelo 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/dg/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='valkey',
        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.valkey7.2.cluster.on'
)

return response

# Creates a cluster mode enabled
response = create_cluster_mode_enabled(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
    ReplicationGroupId='valkey20210',
# 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 in ElastiCache](#).

Crea un cluster abilitato alla modalità cluster con e TLS RBAC

Per garantire la sicurezza, è possibile utilizzare Transport Layer Security (TLS) e Role-Based Access Control (RBAC) durante la creazione di un cluster abilitato alla modalità cluster. A differenza di Valkey o Redis OSSAUTH, in cui tutti i client autenticati hanno 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 degli accessi basato sui ruoli \(\) RBAC](#).

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

```
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.valkey7.2.clu
    """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/dg/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': 'valkey',
            '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='7.2',
        ReplicationGroupDescription='Valkey cluster mode enabled with replicas',
        ReplicationGroupId='valkey2021',
        # 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 in ElastiCache](#).

Controllare che esistano utenti/gruppi di utenti, altrimenti crearli

ConRBAC, si creano utenti e si assegnano 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 ElastiCache gruppi di replica (Redis). OSS In questo modo, è possibile stabilire limiti di sicurezza tra i client che utilizzano lo stesso gruppo o gruppi di OSS replica Valkey o Redis e impedire ai client di accedere ai dati degli altri. Per ulteriori informazioni, consulta [Controllo degli accessi basato sui ruoli \(\) RBAC](#).

Copia il seguente programma e incollalo 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.

Questo esempio utilizza una stringa di accesso con le autorizzazioni per l'utente. Per ulteriori informazioni sulle stringhe di accesso, vedere. [Specifica delle autorizzazioni mediante una stringa di accesso](#)

```
import boto3
import logging

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

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.
    :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
```

Tutorial: Connessione a ElastiCache

I seguenti esempi utilizzano il OSS client Valkey o Redis a cui connettersi. 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

Copiare il seguente programma e incollarlo 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

I seguenti esempi utilizzano boto3 SDK con cui ElastiCache lavorare ElastiCache (Redis). OSS

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 .py. SetAndGetStrings

```
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

Copiate il seguente programma e incollatelo 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

Copiate il seguente programma e incollatelo 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

Copiate il seguente programma e incollatelo 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 Lambda per ElastiCache l'accesso in un 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

- [Passaggio 1: creazione di una cache serverless. ElastiCache](#)
- [Fase 2: Creare una funzione Lambda per ElastiCache](#)
- [Fase 3: testare la funzione Lambda con ElastiCache](#)
- [Fase 4: Pulizia \(opzionale\)](#)

Passaggio 1: creazione di una cache serverless. ElastiCache

Per creare una cache serverless, segui questi passaggi.

Passaggio 1.1: Creare una cache serverless

In questo passaggio, crei una cache serverless nell'Amazon predefinito VPC nella regione us-east-1 del tuo account utilizzando (). AWS Command Line Interface CLI Per informazioni sulla creazione di cache serverless utilizzando la ElastiCache console oppure, consulta. API [Crea una cache serverless Valkey](#)

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

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 (RedisOSS) 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.

Passaggio 1.3: Crea ruolo IAM

1. Crea un documento sulla politica di IAM fiducia, come mostrato di seguito, per il tuo ruolo che consenta al tuo 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 IAM di policy, come illustrato 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 IAM ruolo.

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

4. Crea la IAM politica.

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

5. Allega la IAM politica 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 IAM abilitato.

```
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
```

Fase 2: Creare una funzione Lambda per ElastiCache

Per creare una funzione Lambda per accedere alla ElastiCache cache, procedi nel seguente modo.

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 precedente.

```
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 = urlunparse(
        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.us-east-1.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 Valkey.")
else:
    raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
{decoded_result}")

return "Fetched value from Valkey"
```

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 di autenticazione generati per 15 minuti. IAM 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: Creare il IAM ruolo (ruolo di esecuzione)

Allega la politica AWS gestita denominata `AWSLambdaVPCLambdaAccessExecutionRole` al ruolo.

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

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

In questo passaggio, crei la funzione Lambda (`AccessValkey`) utilizzando il comando AWS CLI `create-function`.

Dalla directory del progetto che contiene il file.zip del pacchetto di distribuzione, esegui il seguente comando CLI `create-function` Lambda.

Per l'opzione `role`, utilizzate il ruolo ARN di esecuzione creato nel passaggio precedente. Per `vpc-config` inserisci gli elenchi separati da virgole delle sottoreti predefinite e VPC dell'ID del gruppo di sicurezza predefinito. VPC Puoi trovare questi valori nella VPC console Amazon. Per trovare le sottoreti predefinite, scegli VPC `YourVPCs`, quindi scegli le impostazioni predefinite del tuo AWS account. VPC Per trovare il gruppo di sicurezza adatto VPC, vai su Sicurezza e scegli Gruppi di sicurezza. Assicurati di aver selezionato la regione `us-east-1`.

```
aws lambda create-function \
  --function-name AccessValkey \
  --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: testare la funzione Lambda con ElastiCache

In questo passaggio, si richiama la funzione Lambda manualmente utilizzando il comando `invoke`. Quando la funzione Lambda viene eseguita, genera un file UUID e lo scrive ElastiCache nella cache specificata nel codice Lambda. Successivamente la funzione Lambda recupera la voce dalla cache.

1. Invocate la funzione Lambda `AccessValkey ()` utilizzando AWS Lambda il comando `invoke`.

```
aws lambda invoke \  
--function-name AccessValkey \  
--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/`). `AccessValkey` Il flusso di log genera un output simile al seguente:

```
Success: Inserted 826e70c5f4d2478c8c18027125a3e01e. Fetched  
826e70c5f4d2478c8c18027125a3e01e from Valkey.
```

- Controlla i risultati nella console. AWS Lambda

Fase 4: Pulizia (opzionale)

Per eseguire la pulizia, procedi nel seguente modo.

Fase 4.1: Eliminare la funzione Lambda

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

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: Rimuovi IAM ruolo e politiche

```
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 ElastiCache cluster

Se hai bisogno di un controllo granulare sul tuo ElastiCache cluster, puoi scegliere di progettare il tuo cluster. ElastiCache consente di gestire un cluster basato su nodi scegliendo il tipo di nodo, il numero di nodi e la posizione 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.

Per informazioni sulla configurazione, consulta [Configurazione ElastiCache](#). Per i dettagli sulla gestione, l'aggiornamento o l'eliminazione di nodi o cluster, consulta [Gestione dei nodi in ElastiCache](#). Per una panoramica dei componenti principali di una ElastiCache distribuzione Amazon quando progetti il tuo ElastiCache cluster, consulta questi [concetti chiave](#).

Argomenti

- [ElastiCache componenti e caratteristiche](#)
- [ElastiCache terminologia](#)
- [Tutorial: Come progettare il proprio cluster](#)
- [Eliminazione di un cluster](#)
- [Altri ElastiCache tutorial e video](#)
- [Gestione dei nodi in ElastiCache](#)
- [Gestione dei cluster in ElastiCache](#)
- [Confronto tra le cache autoprogettate da ValkeyOSS, Redis e Memcached](#)
- [Migrazione online per Valkey o Redis OSS](#)
- [Scelta delle regioni e delle zone di disponibilità per ElastiCache](#)

ElastiCache componenti e caratteristiche

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

Argomenti

- [ElastiCache nodi](#)
- [ElastiCache frammenti](#)
- [ElastiCache cluster](#)

- [ElastiCache replica](#)
- [ElastiCache endpoint](#)
- [ElastiCache gruppi di parametri](#)
- [ElastiCache sicurezza](#)
- [ElastiCache gruppi di sottoreti](#)
- [ElastiCache backup](#)
- [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 dimensioni fisse collegato alla rete. RAM 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 [Ridimensionamento 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 ha il proprio nome e porta Domain Name Service (). DNS 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 una pay-as-you-go base, 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](#).

Per ulteriori informazioni sui nodi, consulta [Gestione dei nodi in ElastiCache](#).

ElastiCache frammenti

Uno OSS shard Valkey o Redis (chiamato gruppo di nodi in API andCLI) è un raggruppamento da uno a sei nodi correlati. Un cluster Valkey o Redis con modalità OSS cluster abilitata 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.

OSSI cluster Valkey e Redis con modalità cluster abilitata possono avere fino a 500 shard, con i dati partizionati tra gli shard. Il limite di nodi o shard può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Valkey o Redis è 5.0.6 o successiva. OSS 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 insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo troppo piccolo o che le sottoreti sono condivise e utilizzate intensamente 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 [Lavorare con i frammenti in ElastiCache](#).

ElastiCache cluster

[Un cluster è un raggruppamento logico di uno o più nodi](#). I dati vengono partizionati tra i nodi di un cluster Memcached e tra gli shard di un cluster Valkey o OSS Redis con la modalità cluster abilitata.

Molte operazioni sono rivolte ai cluster: ElastiCache

- 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 in ElastiCache](#) e [Gestione dei nodi in ElastiCache](#)

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 agli errori dei cluster e dei gruppi di replica Valkey o RedisOSS.

Configurazioni tipiche dei cluster

Di seguito sono riportate le configurazioni tipiche dei cluster.

Cluster Valkey o Redis OSS

OSSI cluster Valkey o Redis con modalità cluster disabilitata contengono sempre un solo shard (al termine, un gruppo di nodi). API CLI Uno OSS shard Valkey o 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 agli errori, consigliamo di avere almeno due nodi in un cluster Valkey o Redis OSS e di abilitare Multi-AZ. Per ulteriori informazioni, consulta [Limitazione dell'impatto degli errori](#).

Se la domanda del OSS cluster Valkey o Redis cambia, puoi scalare verso l'alto o verso il basso. A tale scopo, sposta il cluster in un tipo di istanza di nodo diverso. Se la tua applicazione richiede un uso intensivo di lettura, ti consigliamo di aggiungere repliche di sola lettura al cluster. 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 [Suddivisione dei dati su più livelli in ElastiCache](#).

ElastiCache supporta la modifica dinamica del tipo di nodo di un OSS cluster Valkey o Redis con un tipo di nodo più grande. Per informazioni sul dimensionamento verso l'alto o verso il basso,

consulta [Scalabilità di cluster a nodo singolo per Valkey o Redis \(modalità cluster disabilitata\) OSS](#) o [Scalabilità dei nodi di replica per Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

Configurazioni di cluster tipiche per Memcached

Memcached supporta fino a 300 nodi per cliente per ogni AWS regione, con ogni cluster composto da 1 a 60 nodi. Esegui la partizione dei dati tra i nodi di un cluster Memcached.

Quando si utilizza il motore Memcached, i cluster possono essere composti da 1—60 nodi. Esegui la partizione del database tra i nodi. L'applicazione esegue operazioni di lettura e di scrittura sull'endpoint di ciascun nodo. Per ulteriori informazioni, consulta [Individuazione automatica](#).

Per una migliore tolleranza agli errori, posiziona i nodi Memcached in varie zone di disponibilità (AZs) all'interno della regione del cluster. AWS In questo modo, un guasto in una zona di disponibilità ha un impatto minimo sul cluster e sull'applicazione. Per ulteriori informazioni, consulta [Limitazione dell'impatto degli errori](#).

Quando cambia la domanda per il cluster Memcached, puoi aumentare o diminuire le dimensioni del cluster aggiungendo o rimuovendo nodi che partizionano nuovamente i dati nella nuova quantità di nodi. Ti consigliamo di utilizzare un hashing coerente quando partizioni i dati. Per ulteriori informazioni sull'hashing coerente, consulta [Configurazione ElastiCache del client per un bilanciamento efficiente del carico \(Memcached\)](#).

ElastiCache replica

Per Valkey e RedisOSS, la replica viene implementata raggruppando da due a sei nodi in uno shard (nella forma API eCLI, chiamato gruppo di nodi). Uno di questi nodi è quello di primario di lettura/scrittura. Tutti gli altri sono nodi di replica di sola lettura. Le repliche sono disponibili solo per ElastiCache Valkey e Redis OSS e non per (Memcached). ElastiCache

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 OSS cluster Valkey o Redis supportano uno shard (nella forma e, chiamato gruppo di nodi). API CLI

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

Confronto tra la replica: Valkey o Redis OSS (modalità cluster disabilitata) e Valkey o Redis OSS (modalità cluster abilitata) --> Cluster Valkey o Redis con modalità cluster abilitata rispetto a Valkey o Redis OSS cluster con modalità cluster disabilitata OSS

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

	Cluster Valkey o Redis con modalità cluster disabilitata OSS	Cluster Valkey o Redis con modalità OSS 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
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:

	Cluster Valkey o Redis con modalità cluster disabilitata OSS	Cluster Valkey o Redis con modalità OSS cluster abilitata
--	--	---

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 migrare a Valkey o Redis OSS (modalità cluster abilitata).

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

Tutti gli shard (nei gruppi di nodi API eCLI) 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 OSS 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 consentire letture ElastiCache 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.

ElastiCache endpoint

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

Endpoint a nodo singolo per Valkey o Redis OSS con modalità cluster disabilitata

L'endpoint per un cluster Valkey o Redis a nodo singolo viene utilizzato per connettersi al OSS cluster sia in lettura che in scrittura.

Endpoint multinodo per Valkey o Redis con modalità cluster disabilitata OSS

Un cluster Valkey o Redis a più nodi con modalità OSS cluster disabilitata ha 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. Utilizza i singoli endpoint del nodo per le operazioni di lettura (in API/CLI questi sono indicati come endpoint di lettura).

Endpoint Valkey o Redis OSS (modalità cluster abilitata)

Un cluster Valkey o Redis con modalità OSS cluster abilitata ha un unico 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 [Ricerca degli endpoint di connessione in ElastiCache](#).

ElastiCache Endpoint (Memcached)

Ciascun nodo di un cluster Memcached ha il proprio endpoint. Anche il cluster ha un proprio endpoint, chiamato endpoint di configurazione. Se abiliti la funzione Individuazione automatica e ti connetti all'endpoint di configurazione, la tua applicazione conosce automaticamente l'endpoint di ciascun nodo, anche dopo l'aggiunta o la rimozione di nodi dal cluster. Per ulteriori informazioni, consulta [Individuazione automatica](#).

Per ulteriori informazioni, consulta la pagina relativa agli [endpoint](#).

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 relativi valori predefiniti e di quelli che possono essere modificati, vedere [DescribeEngineDefaultParameters](#) (CLI: [describe-engine-default-parameters](#)).

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

ElastiCache sicurezza

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

Per impostazione predefinita, tutti i nuovi ElastiCache cluster vengono lanciati in un ambiente Amazon Virtual Private Cloud (AmazonVPC). Puoi utilizzare i gruppi di sottoreti per concedere l'accesso ai cluster da EC2 istanze Amazon in esecuzione su sottoreti specifiche.

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

- [Sicurezza dei dati in Amazon ElastiCache](#)
- [Autenticazione con i comandi Valkey e Redis OSS AUTH](#)

ElastiCache gruppi di sottoreti

Un gruppo di sottoreti è una raccolta di sottoreti (in genere private) che puoi designare per i tuoi cluster in esecuzione in un ambiente Amazon VPC.

Se crei un cluster in AmazonVPC, devi specificare un gruppo di sottoreti di 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 di cache in un VPC ambiente Amazon, consulta quanto segue:

- [Amazon VPCs e la ElastiCache sicurezza](#)
- [Fase 3. Autorizza l'accesso al cluster](#)
- [Sottoreti e gruppi di sottoreti](#)

ElastiCache backup

Un backup è una point-in-time copia di un OSS cluster Valkey o Redis o di una cache serverless o di una cache serverless Memcached. 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 Valkey o Redis OSS in esecuzione sul cluster, il processo di backup richiede diverse quantità di memoria riservata per avere successo. 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](#)
- [Assicurarsi di disporre di memoria sufficiente per creare un'istantanea Valkey o Redis OSS](#)

ElastiCache eventi

Quando si verificano eventi importanti su un cluster di cache, ElastiCache invia una notifica a un SNS argomento Amazon specifico. 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 [SNSMonitoraggio degli ElastiCache eventi da Amazon](#).

ElastiCache terminologia

A ottobre 2016, Amazon ElastiCache ha lanciato il supporto per Redis OSS 3.2. A quel punto, abbiamo aggiunto il supporto per il partizionamento dei dati su un massimo di 500 shard (chiamati gruppi di nodi in the and). ElastiCache API AWS CLI Per preservare la compatibilità con le versioni precedenti, abbiamo esteso le operazioni della API versione 2015-02-02 per includere la nuova funzionalità Redis. OSS

Allo stesso tempo, abbiamo iniziato a utilizzare nella ElastiCache console la terminologia utilizzata in questa nuova funzionalità e comune in tutto il settore. Queste modifiche implicano che, in alcuni punti, la terminologia utilizzata nella API console CLI potrebbe essere diversa dalla terminologia utilizzata nella console. L'elenco seguente identifica i termini che potrebbero differire tra API e CLI e console.

Cluster o nodo di cache e nodo

Esiste una one-to-one relazione tra un nodo e un cluster di cache quando non ci sono nodi di replica. Pertanto, la ElastiCache console utilizzava spesso i termini in modo intercambiabile. 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.

ElastiCache APIE AWS CLI continuano a utilizzare i termini come in passato.

Cluster o gruppo di replica Valkey o Redis OSS

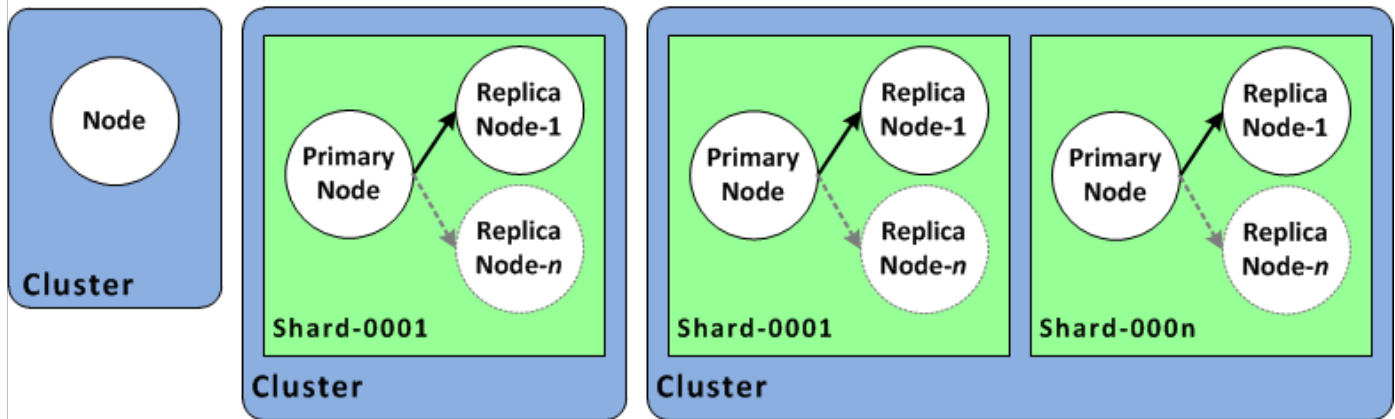
La console ora utilizza il termine cluster per tutti i cluster ElastiCache (Redis). OSS La console utilizza il termine cluster in tutti questi casi:

- Quando il cluster è un cluster Valkey o Redis a nodo singolo. OSS
- Quando il cluster è un cluster Valkey o Redis OSS (modalità cluster disattivata) che supporta la replica all'interno di un singolo shard (nel API gruppo di nodi). CLI
- Quando il cluster è un cluster Valkey o Redis OSS (modalità cluster abilitata) che supporta la replica entro 1-90 shard o fino a 500 con una richiesta di aumento del limite. 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 Valkey o Redis, vedere. OSS [Alta disponibilità utilizzando gruppi di replica](#)

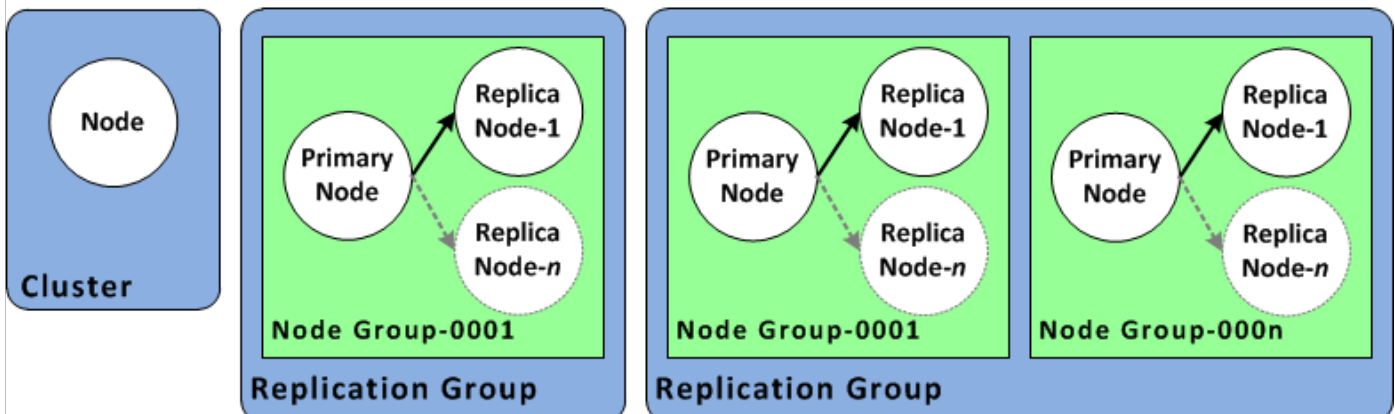
Il diagramma seguente illustra le varie topologie dei cluster ElastiCache (RedisOSS) dal punto di vista della console.

ElastiCache (Redis OSS): Console View



Le AWS CLI operazioni ElastiCache API and distinguono ancora i cluster a nodo singolo ElastiCache (RedisOSS) dai gruppi di replica Valkey o Redis a più nodi. OSS Il diagramma seguente illustra le varie topologie ElastiCache (Redis) dal punto di vista and. OSS ElastiCache API AWS CLI

ElastiCache (Redis OSS): API/CLI View



Gruppo di replica Valkey o Redis OSS e datastore globale

Un datastore globale è una raccolta di uno o più cluster che si replicano tra loro in diverse regioni, mentre un gruppo di replica Valkey o Redis OSS replica i dati su un cluster abilitato alla modalità cluster con più shard. 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 diversa rispetto al cluster primario. AWS

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

Tutorial: Come progettare il proprio cluster

Ecco come progettare il proprio cluster per Valkey e RedisOSS.

Argomenti

- [Progettazione del proprio ElastiCache cluster \(Valkey\)](#)
- [Progettazione del proprio ElastiCache cluster \(RedisOSS\)](#)

Progettazione del proprio ElastiCache cluster (Valkey)

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

Fase 1: creare un gruppo di sottoreti

Prima di creare un cluster ElastiCache (Valkey), è necessario creare un gruppo di sottoreti. Un gruppo di sottoreti di cache è una raccolta di sottoreti che è possibile designare per i cluster di cache in un VPC. Quando si avvia un cluster di cache in un VPC, è necessario selezionare un gruppo di sottoreti di cache. Quindi ElastiCache utilizza quel gruppo di sottoreti di cache per assegnare gli indirizzi IP all'interno di quella sottorete a ciascun nodo di cache del 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, vedere. ElastiCache [Configurazione ElastiCache](#)

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. Accedi alla console di AWS gestione e apri la ElastiCache console 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 VPCID, scegli l'Amazon VPC che hai creato.
 - d. Negli elenchi Zona di disponibilità e ID di sottorete, scegli la Zona di disponibilità o [Utilizzo delle zone locali con ElastiCache](#) l'ID della tua sottorete privata, quindi scegli 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 [↗](#)

i 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-████████		172.31.16.0/20
us-east-1b	subnet-f████████		172.31.32.0/20
us-east-1c	subnet-████████		172.31.0.0/20
us-east-1d	subnet-████████		172.31.80.0/20

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

Il nuovo gruppo di sottoreti viene visualizzato nell'elenco dei gruppi di sottoreti della console.

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 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 in ElastiCache](#). 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. Finché non la elimini, ti verranno addebitati i costi di ElastiCache utilizzo standard per l'istanza. L'addebito totale sarà minimo (in genere meno di un dollaro) se completi l'esercizio descritto qui in una sola seduta ed elimini il cluster alla fine. Per ulteriori informazioni sui tassi di ElastiCache utilizzo, consulta [Amazon ElastiCache](#).

Il cluster viene avviato in un cloud privato virtuale (VPC) basato sul VPC servizio Amazon.

Creazione di un cluster Valkey (modalità cluster disabilitata) (console)

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

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <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 i passaggi descritti in [Creazione di un cloud privato virtuale](#) (). VPC
5. Nella pagina della ElastiCache dashboard, scegli Valkey cache o Redis OSS cache, quindi scegli Crea cache Valkey o Redis cache. OSS
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 Cloud AWS , 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. OSS
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 Valkey e Redis OSS](#) e [Creazione di un gruppo di ElastiCache 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 [Suddivisione dei dati su più livelli in ElastiCache](#).

- 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 entrambi IPv6 gli indirizzi assegnati per funzionare in modalità dual-stack IPv4 e una sottorete -only per funzionare solo come -only. IPv6 IPv6

Quando crei un nuovo gruppo di sottoreti, inserisci l'ID a cui appartiene. VPC

Per ulteriori informazioni, consultare:

- [Scelta del tipo di rete in ElastiCache.](#)
- [Crea una sottorete nel tuo VPC](#)

Se si esegue per [Utilizzo delle zone locali 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:
 - 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 ognle partizioni, seleziona la zona di disponibilità dall'elenco.

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


5. Seleziona Successivo.
6. Nelle impostazioni Advanced Valkey o Redis OSS

- 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 selezionando la chiave gestita dal cliente e scegliendo la AWS KMS chiave. Per ulteriori informazioni, consulta [Utilizzo delle chiavi gestite dai clienti](#) di AWS KMS.

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#). Per Valkey e per la versione 6.0 e successive OSS del motore Redis, se abiliti Encryption in-transit ti verrà 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 la console e CLI](#).
 - AUTHUtente predefinito: un meccanismo di autenticazione per il server Redis. OSS Per ulteriori informazioni, consulta [AUTH](#).
- AUTH— Un meccanismo di autenticazione per il OSS server Redis. Per ulteriori informazioni, vedere [AUTH](#).

 Note

Per Valkey e per OSS le versioni Redis comprese tra 3.2.6 e successive, esclusa la versione 3.2.10, Redis è l'unica opzione. OSS AUTH

- 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. Puoi utilizzare il gruppo di sicurezza predefinito per il tuo o crearne uno nuovo. VPC

Per ulteriori informazioni sui gruppi di sicurezza, consulta [la sezione Gruppi di sicurezza per te VPC](#) nella Amazon VPC User Guide.

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 sul OSS backup e il ripristino di Redis, consulta [Snapshot e ripristino](#).

8. (Facoltativo) Specifica una finestra di manutenzione. La finestra di manutenzione è l'orario, generalmente di un'ora, ogni settimana in cui si ElastiCache pianifica la manutenzione del sistema per il cluster. È possibile ElastiCache scegliere il giorno e l'ora della finestra di manutenzione (nessuna preferenza) oppure scegliere autonomamente il giorno, l'ora e la durata (Specificare la finestra di manutenzione). Se dagli elenchi si sceglie Specify maintenance window (Specificare 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 sono UCT orari.

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

9. (Facoltativo) PerRegistro:
 - In Formato registro, scegli Testo 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 [Taggare le tue risorse ElastiCache](#).
11. Scegli Next (Successivo).
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 ElastiCache](#) (Redis) con Multi-AZ OSS
2. Per completare la creazione del cluster, segui la procedura in [Utilizzo outpost](#).

Non appena lo stato del cluster è disponibile, puoi concedere ad Amazon EC2 l'accesso, connetterti e iniziare a utilizzarlo. Per ulteriori informazioni, consulta [Fase 3. Autorizza l'accesso al cluster](#) e [Fase 4. Connect al 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 in ElastiCache](#).

Creazione di un cluster Valkey (modalità cluster disabilitata) (AWS CLI)

Example

Il CLI codice seguente crea un cluster di cache Valkey (modalità cluster disabilitata) senza repliche.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine valkey \  
--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 valkey ^  
--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 Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#).
- Per utilizzare il AWS CLI, vedere. [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(AWS CLI\)](#)

Fase 3. Autorizza l'accesso al cluster

Questa sezione presuppone che tu abbia dimestichezza con l'avvio e la connessione alle istanze AmazonEC2. Per ulteriori informazioni, consulta la [Amazon EC2 Getting Started Guide](#).

Tutti i ElastiCache cluster sono progettati per essere accessibili da un'EC2istanza Amazon. Lo scenario più comune consiste nell'accedere a un ElastiCache cluster da un'EC2istanza Amazon nello stesso Amazon Virtual Private Cloud (AmazonVPC), come nel caso di questo esercizio.

Per impostazione predefinita, l'accesso di rete al cluster è riservato all'account che è stato utilizzato per avviarlo. Prima di poterti connettere a un cluster da un'EC2istanza, devi autorizzare l'EC2istanza ad accedere al cluster.

Il caso d'uso più comune è quando un'applicazione distribuita su un'EC2istanza deve connettersi a un cluster della stessa VPC. Il modo più semplice per gestire l'accesso tra EC2 istanze e cluster all'interno di uno stesso sistema VPC consiste nel fare quanto segue:

1. Crea un gruppo VPC di sicurezza per il tuo cluster. Questo gruppo di sicurezza può essere utilizzato per limitare l'accesso alle istanze del cluster. Ad esempio, puoi creare una regola personalizzata per questo gruppo di sicurezza che consenta l'accesso TCP utilizzando la porta assegnata al cluster al momento della creazione e un indirizzo IP che utilizzerai per accedere al cluster.

La porta predefinita per i OSS cluster e i gruppi di replica Valkey o Redis è. 6379

Important

I gruppi ElastiCache di sicurezza Amazon sono applicabili 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 gestisci i tuoi ElastiCache nodi in AmazonVPC, controlla l'accesso ai tuoi cluster con i gruppi di VPC sicurezza Amazon, che sono diversi dai gruppi ElastiCache di sicurezza.

Per ulteriori informazioni sull'utilizzo ElastiCache in AmazonVPC, consulta [Amazon VPCs e la ElastiCache sicurezza](#)

2. Crea un gruppo VPC di sicurezza per le tue EC2 istanze (server web e applicativi). Questo gruppo di sicurezza può, se necessario, consentire l'accesso all'EC2istanza da Internet tramite la

tabella VPC di routing dell'istanza. Ad esempio, è possibile impostare regole su questo gruppo di sicurezza per consentire TCP l'accesso all'EC2istanza tramite la porta 22.

3. Crea regole personalizzate nel gruppo di sicurezza per il tuo Cluster che consentano le connessioni dal gruppo di sicurezza che hai creato per le tue EC2 istanze. 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 crei un gruppo di sottoreti in quella zona locale, il tuo VPC viene esteso a quella zona locale e tratterai la sottorete come qualsiasi sottorete in qualsiasi altra zona di disponibilità. VPC Tutti i gateway e le tabelle di routing pertinenti verranno regolati automaticamente.

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

1. Accedi alla Console di AWS gestione e apri la VPC console Amazon all'indirizzo <https://console.aws.amazon.com/vpc>.
2. Fai clic su Security Groups (Gruppi di sicurezza) nel pannello 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. Da Tipo scegli Regola personalizzata TCP.

- a. Per Port Range (Intervallo porte) specificare la porta utilizzata alla creazione del cluster.

La porta predefinita per i OSS cluster e i gruppi di replica Valkey o Redis è. 6379

- b. Nella casella Source (fonte) iniziare a digitare l'ID del gruppo di sicurezza. Dall'elenco seleziona il gruppo di sicurezza che utilizzerai per le tue EC2 istanze Amazon.

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	-

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

Per informazioni sull'accesso al tuo ElastiCache cluster da un'altra AmazonVPC, da una AWS regione diversa o persino dalla tua rete aziendale, consulta quanto segue:

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

Fase 4. Connect al nodo del cluster

Prima di continuare, completa [Fase 3. Autorizza l'accesso al cluster](#).

Questa sezione presuppone che tu abbia creato un'EC2istanza Amazon e che tu possa connetterti ad essa. Per istruzioni su come eseguire questa operazione, consulta la [Amazon EC2 Getting Started Guide](#).

Un'EC2istanza Amazon può connettersi a un nodo del cluster solo se l'hai autorizzata a farlo.

Individuazione degli endpoint dei nodi

Quando il cluster è nello stato disponibile e hai autorizzato l'accesso ad esso, puoi accedere a un'EC2istanza Amazon e connetterti al cluster. A questo scopo, devi innanzitutto determinare l'endpoint.

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

Se un cluster Valkey (modalità cluster disabilitata) ha un solo nodo, l'endpoint del nodo viene utilizzato sia per le letture che per le scritture. 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 DNS nome che si risolve sempre nel nodo primario del 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 suddividerà in modo uniforme le connessioni in entrata all'endpoint tra tutte le repliche di lettura in un cluster. ElastiCache 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. È possibile posizionare le repliche di lettura multiple del ElastiCache cluster in diverse zone di AWS disponibilità (AZ) per garantire l'elevata disponibilità degli endpoint di lettura.

Note

Un endpoint di lettura non è un servizio di load balancer. È un DNS record che verrà risolto 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 trovare gli endpoint di un cluster Valkey (modalità cluster disabilitata)

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

Apparirà la schermata dei cluster con un elenco che includerà tutte le cache OSS serverless Valkey o Redis esistenti, i cluster Valkey (modalità cluster disabilitata) e Valkey (modalità cluster abilitata). Scegliere il cluster creato nella sezione [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#).

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 [redacted]	Description [redacted]	Node type cache.r6g.large	Status Available
Engine Redis OSS	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 primari e Reader per un cluster Valkey (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 Valkey (modalità cluster disabilitata) dispone di nodi di replica, è possibile trovare gli endpoint del nodo di replica del cluster scegliendo il nome del cluster e quindi selezionando la scheda 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	redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-002	available	replica	6379	redis-01.7abc2d.0002.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-003	available	replica	6379	redis-01.7abc2d.0003.usw2.cache.amazonaws.com:6379

Endpoint del nodo per un cluster Valkey (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, consulta [Connessione ai nodi](#).

Un endpoint primario Valkey (modalità cluster disabilitata) ha un aspetto simile al seguente. 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.

- [Ricerca degli endpoint di connessione in ElastiCache](#)

- [Ricerca degli endpoint per un cluster \(console\) Valkey o Redis OSS \(modalità cluster abilitata\)](#): è necessario l'endpoint di configurazione del cluster.
- [Individuazione degli endpoint \(AWS CLI\)](#)
- [Ricerca degli endpoint \(\) ElastiCache API](#)

Connect a un OSS cluster o gruppo di replica Valkey o Redis (Linux)

Ora che hai l'endpoint di cui hai bisogno, puoi accedere a un'EC2istanza e connetterti al cluster o al gruppo di replica. Nell'esempio seguente, si utilizza l'utilità valkey-cli per connettersi a un cluster. L'ultima versione di valkey-cli supporta anche/per la connessione di cluster abilitati alla crittografia/autenticazione. SSL TLS

L'esempio seguente utilizza EC2 istanze Amazon che eseguono Amazon Linux e Amazon Linux 2. Per i dettagli sull'installazione e la compilazione di valkey-cli con altre distribuzioni Linux, consulta la documentazione del tuo sistema operativo specifico.

Note

Questo processo riguarda il test di una connessione utilizzando l'utilità valkey-cli solo per uso non pianificato. [Per un elenco dei client Valkey e Redis OSS supportati, consulta la documentazione di Valkey.](#) Per esempi di utilizzo di AWS SDKs with ElastiCache, consulta [Tutorial: Guida introduttiva a Python e ElastiCache](#)

Connessione a un cluster non crittografato disattivato in modalità cluster

1. Esegui il comando seguente per connetterti al cluster e sostituirlo *primary-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Valkey o Redis OSS è 6379.)

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

Il risultato in un prompt dei OSS comandi di Valkey o Redis è simile al seguente:

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

2. È ora possibile eseguire i comandi Valkey o Redis. OSS

```
set x Hello
OK

get x
"Hello"
```

Connessione a un cluster non crittografato attivato in modalità cluster

1. Esegui il comando seguente per connetterti al cluster e sostituirlo *configuration-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Valkey o Redis OSS è 6379.)

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

Note

Nel comando precedente, l'opzione -c abilita la modalità cluster seguendo - e - reindirizzamenti. ASK MOVED

Il risultato in un prompt dei OSS comandi di Valkey o Redis è simile al seguente:

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

2. È ora possibile eseguire i comandi Valkey o Redis. OSS 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. MOVED Per ulteriori informazioni sull'MOVEDerrore, consulta le specifiche del [OSScluster 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, valkey-cli utilizza una connessione non crittografata per la TCP connessione a Valkey o Redis. OSS L'opzione BUILD_TLS=yes abilita SSL/TLS al momento della compilazione di valkey-cli, come mostrato nella sezione precedente. [Scarica e configura l'accesso alla riga di comando](#) L'abilitazione è facoltativa. AUTH Tuttavia, è necessario abilitare la crittografia in transito per abilitare. AUTH Per ulteriori dettagli sulla ElastiCache crittografia e l'autenticazione, vedere [ElastiCache crittografia in transito \(\) TLS](#).

Note

È possibile utilizzare l'opzione `--tls` con valkey-cli per connettersi a cluster crittografati abilitati e disabilitati in modalità cluster. Se un cluster ha un set di AUTH token, puoi utilizzare l'opzione `-a` per fornire una password. AUTH

Negli esempi seguenti, assicurati di sostituire *cluster-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Valkey o Redis OSS 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/valkey-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/valkey-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/valkey-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/valkey-cli -c -h cluster-endpoint --tls -p port number
```

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

alternativa a valkey-cli

Se il cluster non è abilitato in modalità cluster ed è necessario stabilire una connessione al cluster per un breve test ma senza eseguire la compilazione valkey-cli, è possibile utilizzare telnet o openssl. Nei seguenti comandi di esempio, assicurati di sostituire *cluster-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Valkey o Redis OSS 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, sostituisci *your-password* con la password per il tuo 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
```

Connect a un OSS cluster o gruppo di replica Valkey o Redis (Windows)

Per connettersi al OSS cluster Valkey o Redis da un'istanza EC2 Windows utilizzando Valkey o Redis OSSCLI, è necessario scaricare il pacchetto valkey-cli e utilizzare valkey-cli.exe per connettersi al cluster Valkey CLI o Redis da un'istanza Windows. OSS EC2

Nell'esempio seguente, si utilizza l'utilità valkey-cli per connettersi a un cluster che non è abilitato alla crittografia e che esegue Valkey o Redis. OSS Per ulteriori informazioni su Valkey o Redis OSS e sui comandi disponibili, consulta Comandi Valkey e Redis sul sito Web di [Valkey](#). OSS

Per connettersi a un cluster Valkey o Redis che non è OSS abilitato alla crittografia utilizzando `valkey-cli`

1. Connettiti alla tua EC2 istanza Amazon utilizzando l'utilità di connessione che preferisci. Per istruzioni su come connettersi a un'EC2 istanza Amazon, consulta la [Amazon EC2 Getting Started Guide](#).
2. Copia e incolla il link <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 Valkey dalla versione disponibile all'indirizzo GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Estrarre il file zip nella cartella/percorso desiderato.

Apri il prompt dei comandi, passa alla directory Valkey ed esegui il comando. c :

```
\Valkey>valkey-cli -h Redis_Cluster_Endpoint -p 6379
```

Per esempio:

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

3. Esegui i comandi Valkey o Redis. OSS

Ora sei connesso al cluster e puoi eseguire OSS comandi Valkey o Redis come i seguenti.

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

A questo punto come si può procedere?

Ora che hai provato l'esercizio Getting Started, puoi esplorare le seguenti sezioni per saperne di più sugli strumenti ElastiCache disponibili:

- [Guida introduttiva con AWS](#)
- [Strumenti per Amazon Web Services](#)
- [Interfaccia a riga di comando AWS](#)
- [ElastiCache API](#)[Riferimento Amazon](#)

Dopo aver completato l'esercizio Getting Started, puoi leggere queste sezioni per saperne di più sull'ElastiCache amministrazione:

- [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.

- [ElastiCache migliori pratiche e strategie di caching](#)

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

Progettazione del proprio ElastiCache cluster (RedisOSS)

Di seguito sono riportate le azioni una tantum da intraprendere per progettare il proprio cluster ElastiCache (RedisOSS).

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

Argomenti

- [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](#)

Fase 1: creare un gruppo di sottoreti

Prima di creare il cluster, assicurati di creare un gruppo di sottoreti. Un gruppo di sottoreti di cache è una raccolta di sottoreti che potresti voler designare per i tuoi cluster di cache in un VPC. Quando si avvia un cluster di cache in un VPC, è necessario selezionare un gruppo di sottoreti di cache. Quindi ElastiCache utilizza quel gruppo di sottoreti di cache per assegnare gli indirizzi IP all'interno di quella sottorete a ciascun nodo di cache del 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.

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. Accedi alla console di AWS gestione e apri la console all' ElastiCache 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 VPCID, scegli l'Amazon VPC che hai 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-		172.31.16.0/20
us-east-1b	subnet-		172.31.32.0/20
us-east-1c	subnet-		172.31.0.0/20
us-east-1d	subnet-		172.31.80.0/20

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

Il nuovo gruppo di sottoreti viene visualizzato nell'elenco dei gruppi di sottoreti della console. 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 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 in ElastiCache](#). 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. Finché non la elimini, ti verranno addebitati i costi di ElastiCache utilizzo standard per l'istanza. L'addebito totale sarà minimo (in genere meno di un dollaro) se completi l'esercizio descritto qui in una sola seduta ed

elimini il cluster alla fine. Per ulteriori informazioni sui tassi di ElastiCache utilizzo, consulta [Amazon ElastiCache](#).

Il cluster viene avviato in un cloud privato virtuale (VPC) basato sul VPC servizio Amazon.

Creazione di un cluster Redis OSS (modalità cluster disabilitata) (console)


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

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <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 i passaggi descritti in [Creazione di un cloud privato virtuale](#) (). VPC
5. Nella pagina della ElastiCache dashboard, scegli Valkey cache o Redis OSS cache. Per questo esercizio sceglieremo Redis OSS cache, quindi sceglieremo Crea cache Redis. OSS
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 Cloud AWS , 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 \(RedisOSS\)](#) 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 Valkey e Redis OSS](#) e [Creazione di un gruppo di ElastiCache 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 [Suddivisione dei dati su più livelli in ElastiCache](#).

- 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 entrambi IPv6 gli indirizzi assegnati per funzionare in modalità dual-stack IPv4 e una sottorete -only per funzionare solo come -only. IPv6 IPv6

Quando crei un nuovo gruppo di sottoreti, inserisci l'ID a cui appartiene. VPC

Per ulteriori informazioni, consultare:

- [Scelta del tipo di rete in ElastiCache.](#)
- [Crea una sottorete nel tuo VPC.](#)

Se si esegue per [Utilizzo delle zone locali 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:

- 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 delle regioni e delle zone di disponibilità per ElastiCache](#).

5. Seleziona Successivo.

6. In Impostazioni Redis OSS avanzate

- 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 gestita AWS KMS dal cliente e scegliendo la chiave. Per ulteriori informazioni, consulta [Utilizzo delle chiavi gestite dai clienti](#) di AWS KMS.

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in](#)

[transito](#). Per la versione 6.0 e successive OSS del motore Redis, se abiliti la crittografia in transito, ti verrà 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 la console e CLI](#).
- AUTHUtente predefinito: un meccanismo di autenticazione per Valkey e server Redis. OSS Per ulteriori informazioni, vedere [AUTH](#)
- AUTH— Un meccanismo di autenticazione per il OSS server Redis. Per ulteriori informazioni, vedere [AUTH](#).

 Note

Per OSS le versioni Redis comprese tra la 3.2.6 e successive, esclusa la versione 3.2.10, Redis OSS 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. Puoi utilizzare il gruppo di sicurezza predefinito per il tuo VPC o crearne uno nuovo.

Per ulteriori informazioni sui gruppi di sicurezza, consulta [la sezione Gruppi di sicurezza per te VPC](#) nella Amazon VPC User Guide.

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, consulta [Snapshot e ripristino](#).

8. (Facoltativo) Specifica una finestra di manutenzione. La finestra di manutenzione è l'orario, generalmente di un'ora, ogni settimana in cui si ElastiCache pianifica la manutenzione del sistema per il cluster. È possibile ElastiCache scegliere il giorno e l'ora

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

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

9. (Facoltativo) PerRegistro:

- In Formato registro, scegli Testo 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 [Taggare le tue risorse ElastiCache](#).
11. Scegli Next (Successivo).
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 ElastiCache](#) (Redis) con Multi-AZ OSS
2. Per completare la creazione del cluster, segui la procedura in [Utilizzo outpost](#).

Non appena lo stato del cluster è disponibile, puoi concedere ad Amazon EC2 l'accesso, connetterti e iniziare a usarlo. Per ulteriori informazioni, consulta [Fase 3. Autorizza l'accesso al cluster](#) e [Fase 4. Connect al 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 in ElastiCache](#).

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

Il CLI codice seguente crea un cluster di cache Redis OSS (modalità cluster disabilitata) 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 Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#).
- Per utilizzare il AWS CLI, vedi. [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(AWS CLI\)](#)

Fase 3: autorizzazione dell'accesso al cluster

Questa sezione presuppone che tu abbia familiarità con l'avvio e la connessione alle istanze AmazonEC2. Per ulteriori informazioni, consulta la [Amazon EC2 Getting Started Guide](#).

Tutti i ElastiCache cluster sono progettati per essere accessibili da un'EC2istanza Amazon. Lo scenario più comune consiste nell'accedere a un ElastiCache cluster da un'EC2istanza Amazon nello stesso Amazon Virtual Private Cloud (AmazonVPC), come nel caso di questo esercizio.

Per impostazione predefinita, l'accesso di rete al cluster è riservato all'account che è stato utilizzato per avviarlo. Prima di poterti connettere a un cluster da un'EC2istanza, devi autorizzare l'EC2istanza ad accedere al cluster. I passaggi richiesti dipendono dal fatto che il cluster sia stato avviato in EC2 - VPC o in EC2 -Classic.

Il caso d'uso più comune è quando un'applicazione distribuita su un'EC2istanza deve connettersi a un cluster della stessa VPC. Il modo più semplice per gestire l'accesso tra EC2 istanze e cluster nello stesso ambiente VPC consiste nel fare quanto segue:

1. Crea un gruppo VPC di sicurezza per il tuo cluster. Questo gruppo di sicurezza può essere utilizzato per limitare l'accesso alle istanze del cluster. Ad esempio, puoi creare una regola personalizzata per questo gruppo di sicurezza che consenta l'accesso TCP utilizzando la porta assegnata al cluster al momento della creazione e un indirizzo IP che utilizzerai per accedere al cluster.

La porta predefinita per i OSS cluster e i gruppi di replica Redis è. 6379

Important

I gruppi ElastiCache di sicurezza Amazon sono applicabili 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 gestisci i tuoi ElastiCache nodi in AmazonVPC, controlla l'accesso ai tuoi cluster con i gruppi di VPC sicurezza Amazon, che sono diversi dai gruppi ElastiCache di sicurezza.

Per ulteriori informazioni sull'utilizzo ElastiCache in AmazonVPC, consulta [Amazon VPCs e la ElastiCache sicurezza](#)

2. Crea un gruppo VPC di sicurezza per le tue EC2 istanze (server web e applicativi). Questo gruppo di sicurezza può, se necessario, consentire l'accesso all'EC2istanza da Internet tramite la

tabella VPC di routing dell'istanza. Ad esempio, è possibile impostare regole su questo gruppo di sicurezza per consentire TCP l'accesso all'EC2istanza tramite la porta 22.

3. Crea regole personalizzate nel gruppo di sicurezza per il tuo Cluster che consentano le connessioni dal gruppo di sicurezza che hai creato per le tue EC2 istanze. Ciò consente a qualsiasi membro del gruppo di sicurezza di accedere ai cluster.

Note

Se hai intenzione di utilizzarle [Utilizzo delle zone locali con ElastiCache](#), assicurati di averle abilitate. Quando crei un gruppo di sottoreti in quella zona locale, il tuo VPC viene esteso a quella zona locale e VPC tratterai 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 una regola in un gruppo di VPC sicurezza che consenta le connessioni da un altro gruppo di sicurezza

1. Accedi alla Console di AWS gestione e apri la VPC console Amazon all'indirizzo <https://console.aws.amazon.com/vpc>.
2. Fai clic su Security Groups (Gruppi di sicurezza) nel pannello 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.

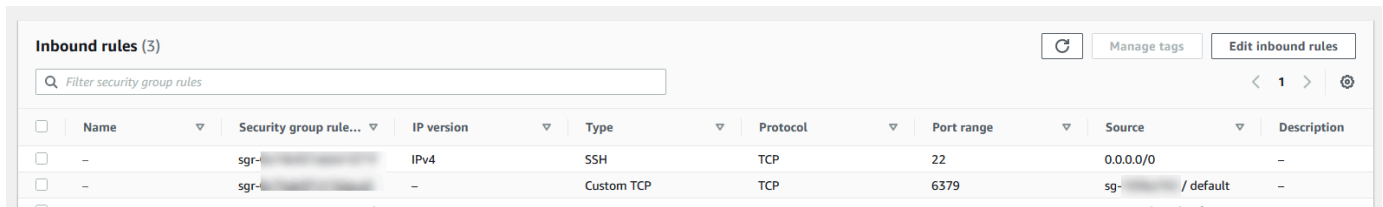
4. Da Tipo scegli Regola personalizzata TCP.

- a. Per Port Range (Intervallo porte) specificare la porta utilizzata alla creazione del cluster.

La porta predefinita per i OSS cluster e i gruppi di replica Redis è. 6379

- b. Nella casella Source (fonte) iniziare a digitare l'ID del gruppo di sicurezza. Dall'elenco seleziona il gruppo di sicurezza che utilizzerai per le tue EC2 istanze Amazon.

5. Scegliere Save (Salva) al termine.



Inbound rules (3)

Filter security group rules

<input type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-...	IPv4	SSH	TCP	22	0.0.0.0/0	-
<input type="checkbox"/>	-	sgr-...	-	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 tuo ElastiCache cluster da un'altra AmazonVPC, da una AWS regione diversa o persino dalla tua rete aziendale, consulta quanto segue:

- [Modelli di accesso per accedere a una ElastiCache cache in Amazon VPC](#)
- [Accesso alle ElastiCache risorse 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 tu abbia creato un'EC2istanza Amazon e che tu possa connetterti ad essa. Per istruzioni su come eseguire questa operazione, consulta la [Amazon EC2 Getting Started Guide](#).

Un'EC2istanza Amazon può connettersi a un nodo del cluster solo se l'hai autorizzata a farlo.

Individuazione degli endpoint dei nodi

Quando il cluster è nello stato disponibile e hai autorizzato l'accesso ad esso, puoi accedere a un'EC2istanza Amazon e connetterti al cluster. A questo scopo, devi innanzitutto determinare l'endpoint.

Ricerca degli endpoint (console) di un cluster Valkey o Redis OSS (modalità cluster disabilitata)

Se un cluster Redis OSS (modalità cluster disabilitata) ha un solo nodo, l'endpoint del nodo viene utilizzato sia per le letture che per le scritture. 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 DNS nome che si risolve sempre nel nodo primario del 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 suddividerà in modo uniforme le connessioni in entrata all'endpoint tra tutte le repliche di lettura in un cluster (Redis). ElastiCache OSS 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. È possibile posizionare le repliche di lettura multiple del cluster ElastiCache (RedisOSS) in diverse zone di disponibilità (AZ) per garantire l'elevata AWS disponibilità degli endpoint di lettura.

Note

Un endpoint di lettura non è un servizio di load balancer. È un DNS record che verrà risolto 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 trovare gli endpoint di un cluster Redis OSS (modalità cluster disabilitata)

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

Apparirà la schermata dei cluster con un elenco che includerà tutte le cache OSS serverless Valkey o Redis esistenti, i cluster Redis (modalità cluster disabilitata) e i cluster Redis OSS (modalità cluster abilitata). OSS Scegliere il cluster creato nella sezione [Creazione di un cluster Redis OSS \(modalità cluster disabilitata\) \(console\)](#).

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 [redacted]	Description [redacted]	Node type cache.r6g.large	Status Available
Engine Redis OSS	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 primari e Reader per un cluster Redis OSS (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 OSS (modalità cluster disabilitata) dispone di nodi di replica, puoi trovare gli endpoint del nodo di replica del cluster scegliendo il nome del cluster e quindi selezionando la scheda 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	redis-01.7abc2d.0001.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-002	available	replica	6379	redis-01.7abc2d.0002.usw2.cache.amazonaws.com:6379
<input type="checkbox"/>	test-no-003	available	replica	6379	redis-01.7abc2d.0003.usw2.cache.amazonaws.com:6379

Endpoint del nodo per un cluster Redis OSS (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, consulta [Connessione ai nodi](#).

Un endpoint primario Redis OSS (modalità cluster disabilitata) ha un aspetto simile al seguente. 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.

- [Ricerca degli endpoint di connessione in ElastiCache](#)

- [Ricerca degli endpoint per un cluster \(console\) Valkey o Redis OSS \(modalità cluster abilitata\)](#): è necessario l'endpoint di configurazione del cluster.
- [Individuazione degli endpoint \(AWS CLI\)](#)
- [Ricerca degli endpoint \(\) ElastiCache API](#)

Connect a un OSS cluster o gruppo di replica Valkey o Redis (Linux)

Ora che hai l'endpoint di cui hai bisogno, puoi accedere a un'EC2istanza e connetterti al cluster o al gruppo di replica. Nell'esempio seguente, si utilizza l'utilità `valkey-cli` per connettersi a un cluster. L'ultima versione di `valkey-cli` supporta anche/per la connessione di cluster abilitati alla crittografia/autenticazione. SSL TLS

L'esempio seguente utilizza EC2 istanze Amazon che eseguono Amazon Linux e Amazon Linux 2. Per i dettagli sull'installazione e la compilazione di `valkey-cli` con altre distribuzioni Linux, consulta la documentazione del tuo sistema operativo specifico.

Note

Questo processo riguarda il test di una connessione utilizzando l'utilità `valkey-cli` solo per uso non pianificato. [Per un elenco dei client supportati, consulta la documentazione di Valkey](#). Per esempi di utilizzo di AWS SDKs with ElastiCache, vedere [Tutorial: Guida introduttiva a Python e ElastiCache](#).

Connessione a un cluster non crittografato disattivato in modalità cluster

1. Esegui il comando seguente per connetterti al cluster e sostituirlo *primary-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Valkey e Redis OSS è 6379.)

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

Il risultato in un prompt dei comandi è simile al seguente:

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

2. È ora possibile eseguire i comandi Valkey e RedisOSS.


```
set x Hello
OK

get x
"Hello"
```

Connessione a un cluster non crittografato attivato in modalità cluster

1. Esegui il comando seguente per connetterti al cluster e sostituirlo *configuration-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Valkey e Redis OSS è 6379.)

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

Note

Nel comando precedente, l'opzione -c abilita la modalità cluster seguendo - e - reindirizzamenti. ASK MOVED

Il risultato in un prompt dei comandi è simile al seguente:

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

2. È ora possibile eseguire i comandi Valkey e RedisOSS. 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. MOVED Per ulteriori informazioni sull'MOVEDerrore, vedere le [specifiche del cluster](#).

```
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, valkey-cli utilizza una connessione non crittografata per la TCP connessione a Valkey e Redis. OSS L'opzione BUILD_TLS=yes abilita SSL/TLS al momento della compilazione di valkey-cli, come mostrato nella sezione precedente. [Scarica e configura l'accesso alla riga di comando](#) L'abilitazione è facoltativa. AUTH Tuttavia, è necessario abilitare la crittografia in transito per abilitare AUTH. Per ulteriori dettagli sulla ElastiCache crittografia e l'autenticazione, vedere [ElastiCache crittografia in transito \(\) TLS](#).

Note

È possibile utilizzare l'opzione `--tls` con valkey-cli per connettersi a cluster crittografati abilitati e disabilitati in modalità cluster. Se un cluster ha un set di AUTH token, puoi utilizzare l'opzione `-a` per fornire una password. AUTH

Negli esempi seguenti, assicurati di sostituire *cluster-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Redis OSS 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/valkey-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/valkey-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/valkey-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/valkey-cli -c -h cluster-endpoint --tls -p port number
```

Dopo esserti connesso al cluster, puoi eseguire i OSS comandi Valkey o Redis come mostrato negli esempi precedenti per i cluster non crittografati.

Un'alternativa a valkey-cli o 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 eseguire la compilazione valkey-cli o redis-cli, è possibile utilizzare telnet o openssl. Nei seguenti comandi di esempio, assicurati di sostituire *cluster-endpoint* e *port number* con l'endpoint del cluster e il numero di porta. (La porta predefinita per Redis OSS 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, sostituisci *your-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
```

Connect a un OSS cluster o gruppo di replica Valkey o Redis (Windows)

Per connettersi al cluster da un'istanza EC2 Windows utilizzando Valkey o Redis OSSCLI, è necessario scaricare il pacchetto valkey-cli e utilizzare valkey-cli.exe per connettersi al cluster Valkey o Redis da un'istanza Windows. OSS EC2

Nell'esempio seguente, si utilizza l'utilità valkey-cli per connettersi a un cluster che non è abilitato alla crittografia e che esegue Valkey o Redis. OSS [Per ulteriori informazioni su Valkey e sui comandi disponibili, vedere Comandi Valkey sul sito Web di Valkey.](#)

Per connettersi a un OSS cluster Valkey o Redis che non è abilitato alla crittografia utilizzando `valkey-cli`

1. Connettiti alla tua EC2 istanza Amazon utilizzando l'utilità di connessione che preferisci. Per istruzioni su come connettersi a un'EC2 istanza Amazon, consulta la [Amazon EC2 Getting Started Guide](#).
2. Copia e incolla il link <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 OSS client Redis dalla versione disponibile all'indirizzo GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Estrarre il file zip nella cartella/percorso desiderato.

Apri il prompt dei comandi, passa alla directory Valkey ed esegui il comando. c :

```
\Valkey>valkey-cli -h Valkey_Cluster_Endpoint -p 6379
```

Per esempio:

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

3. Esegui i comandi Valkey o Redis. OSS

Ora sei connesso al cluster e puoi eseguire OSS comandi Valkey o Redis come i seguenti.

```
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 valkey-cli
```

Eliminazione di un cluster

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

Warning

- Quando si elimina un ElastiCache cluster, le istantanee manuali vengono conservate. 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](#).
- `CreateSnapshot` è necessaria l'autorizzazione per creare un'istantanea finale. Senza questa autorizzazione, la API chiamata avrà esito negativo con un'Access Denied eccezione.

Usando il AWS Management Console

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

Per eliminare un cluster

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nella dashboard del ElastiCache motore, scegli Valkey o RedisOSS.

Viene visualizzato un elenco di tutte le cache in esecuzione su quel motore.

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

È possibile eliminare solo un cluster alla volta dalla ElastiCache console. 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.

Utilizzando il 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 `delete-cache-cluster` elimina solo un 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, vedere l' ElastiCache argomento AWS CLI [delete-cache-cluster](#) per.

Altri ElastiCache tutorial e video

I seguenti tutorial riguardano le attività che interessano l'utente Amazon ElastiCache .

- [ElastiCache Video](#)

- [Tutorial: Configurazione di una funzione Lambda per accedere ad Amazon da ElastiCache Amazon VPC](#)

ElastiCache Video

Ecco alcuni video per aiutarti a imparare i ElastiCache concetti di base e avanzati di Amazon. Per informazioni sulla AWS formazione, consulta [AWS Formazione e certificazione](#).

Argomenti

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

Video introduttivi

I seguenti video ti presentano Amazon ElastiCache.

Argomenti

- [AWS re:Invent 2020: le novità di Amazon ElastiCache](#)
- [AWS re:Invent 2019: le novità di Amazon ElastiCache](#)
- [AWS re:Invent 2017: le novità di Amazon ElastiCache](#)
- [DAT204 — Creazione di applicazioni scalabili senza servizi \(re:Invent 2015 AWS \) SQL](#)
- [DAT207—Accelerazione delle prestazioni delle applicazioni con Amazon ElastiCache \(AWS re:Invent 2013\)](#)

AWS re:Invent 2020: le novità di Amazon ElastiCache

[AWS re:Invent 2020: le novità di Amazon ElastiCache](#)

AWS re:Invent 2019: le novità di Amazon ElastiCache

[AWS re:Invent 2019: le novità di Amazon ElastiCache](#)

AWS re:Invent 2017: le novità di Amazon ElastiCache

[AWS re:Invent 2017: le novità di Amazon ElastiCache](#)

DAT204 — Creazione di applicazioni scalabili senza servizi (re:Invent 2015 AWS) SQL

In questa sessione, discutiamo dei vantaggi dei SQL database No e facciamo un tour dei principali SQL servizi No offerti da AWS—Amazon DynamoDB e Amazon. ElastiCache Poi, ascoltiamo due clienti importanti, Expedia e Mapbox, sui loro casi d'uso e sulle sfide architettoniche e su come li hanno affrontati utilizzando SQL i servizi AWS No, inclusi modelli di progettazione e best practice. Al

termine di questa sessione dovreste avere una migliore conoscenza di No SQL e delle sue potenti funzionalità, pronti ad affrontare le sfide dei database con sicurezza.

[DAT204—Creazione di applicazioni scalabili AWS senza SQL servizi \(re:Invent 2015\)](#)

DAT207—Accelerazione delle prestazioni delle applicazioni con Amazon ElastiCache (AWS re:Invent 2013)

In questo video, scopri come usare Amazon ElastiCache per implementare facilmente un sistema di caching in memoria per accelerare le prestazioni delle tue applicazioni. Ti mostriamo come usare Amazon ElastiCache per migliorare la latenza delle applicazioni e ridurre il carico sui tuoi server di 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 vari scenari e casi d'uso che possono trarre vantaggio dall'abilitazione della memorizzazione nella cache e discuteremo delle funzionalità fornite da Amazon ElastiCache.

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

Video avanzati

I seguenti video trattano ElastiCache argomenti Amazon più avanzati.

Argomenti

- [Progetta per il successo con le ElastiCache best practice di Amazon \(re:Invent 2020\)](#)
- [Potenzia le tue app in tempo reale con Amazon ElastiCache \(re:Invent 2019\)](#)
- [Best practice: migrazione di OSS cluster Redis da Amazon EC2 a ElastiCache \(re:Invent 2019\)](#)
- [Scalabilità di una piattaforma di Fantasy Sports con Amazon ElastiCache e Amazon STP11 Aurora \(re:Invent 2018\)](#)
- [Redis affidabile e scalabile OSS nel cloud con Amazon ElastiCache \(re:Invent 2018\)](#)
- [ElastiCache Deep Dive: modelli di progettazione per archivi di dati in memoria \(re:Invent 2018\)](#)
- [DAT305—Amazon ElastiCache Deep Dive \(re:Invent 2017\)](#)
- [DAT3 ElastiCache 06—Approfondimento su Amazon \(re:Invent 2016\)](#)
- [DAT317—Come IFTTT utilizza ElastiCache \(Redis\) per prevedere gli eventi \(re:Invent OSS 2016\)](#)
- [DAT407—Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)
- [SDD402—Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)
- [DAT307 — Approfondimento ElastiCache sull'architettura e sui modelli di progettazione di Amazon \(re:Invent 2013\)](#)

Progetta per il successo con le ElastiCache best practice di Amazon (re:Invent 2020)

Con la crescita esplosiva di applicazioni aziendali critiche e in tempo reale basate su Redis, la disponibilità OSS, la scalabilità e la sicurezza sono diventate le principali considerazioni. Scopri le best practice per configurare Amazon ElastiCache con successo grazie alla scalabilità online, all'elevata disponibilità nelle implementazioni Multi-AZ e alle configurazioni di sicurezza.

[Progetta per il successo con le ElastiCache best practice di Amazon \(re:Invent 2020\)](#)

Potenzia le tue app 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 aumentare le prestazioni delle applicazioni in tempo reale utilizzando Amazon in memoria completamente gestito ElastiCache per prestazioni estreme, elevata scalabilità, disponibilità e sicurezza.

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

Best practice: migrazione di OSS cluster Redis da Amazon EC2 a ElastiCache (re:Invent 2019)

Gestire i OSS cluster 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 nuova funzionalità di migrazione online per Amazon ElastiCache, ora puoi spostare facilmente i tuoi dati da Redis self-hosted su Amazon EC2 a OSS Amazon completamente gestito ElastiCache, con la modalità cluster disabilitata. In questa sessione, imparerai a conoscere il nuovo strumento di migrazione online, vedrai una demo e, soprattutto, imparerai le best practice pratiche per una migrazione senza problemi ad Amazon. ElastiCache

[Best practice: migrazione di OSS cluster Redis da Amazon EC2 a ElastiCache \(re:Invent 2019\)](#)

Scalabilità di una piattaforma di Fantasy Sports con Amazon ElastiCache e Amazon STP11 Aurora (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, CTO Dream11 Amit Sharma spiega come l'azienda utilizza Amazon Aurora e Amazon ElastiCache per gestire il traffico flash, che può triplicare entro una finestra di risposta di 30 secondi. Sharma parla anche di dimensionare le transazioni senza bloccare e condivide 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 sport di fantasia con Amazon e Amazon ElastiCache Aurora () STP11

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

Redis affidabile e scalabile OSS nel cloud con Amazon ElastiCache (re:Invent 2018)

Questa sessione illustra le funzionalità e i miglioramenti del nostro OSS servizio compatibile con Redis, Amazon ElastiCache (Redis). OSS Tratteremo funzionalità chiave, come Redis OSS 5, miglioramenti della scalabilità e delle prestazioni, sicurezza e conformità e molto altro. Discutiamo anche le funzionalità future e i casi di studio dei clienti.

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

ElastiCache Deep Dive: modelli di progettazione per archivi di dati in memoria (re:Invent 2018)

In questa sessione, diamo un'occhiata al dietro le quinte per conoscere il design e l'architettura di Amazon ElastiCache. Scopri i modelli di progettazione comuni delle nostre offerte Redis OSS e Memcached e come i clienti li utilizzano per l'elaborazione dei dati in memoria per ridurre la latenza e migliorare la produttività delle applicazioni. Esaminiamo le ElastiCache migliori pratiche, i modelli di progettazione e gli anti-pattern.

[ElastiCache Deep Dive: modelli di progettazione per archivi di dati in memoria \(re:Invent 2018\)](#)

DAT305—Amazon ElastiCache Deep Dive (re:Invent 2017)

Guarda dietro le quinte per scoprire il design e l'architettura ElastiCache di Amazon. Scopri i modelli di progettazione comuni delle nostre OSS offerte Memcached e Redis e come i clienti li hanno utilizzati per le operazioni in memoria per ridurre la latenza e migliorare la produttività delle applicazioni. In questo video, esaminiamo le ElastiCache best practices, i modelli di progettazione e gli anti-pattern.

Il video introduce quanto segue:

- ElastiCache (RedisOSS) ricondivisione online
- ElastiCache sicurezza e crittografia

- ElastiCache (RedisOSS) versione 3.2.10

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

DAT3 ElastiCache 06—Approfondimento su Amazon (re:Invent 2016)

Guarda dietro le quinte per scoprire il design e l'architettura ElastiCache di Amazon. Scopri i modelli di progettazione comuni delle nostre OSS offerte Memcached e Redis e come i clienti li hanno utilizzati per le operazioni in memoria per ridurre la latenza e migliorare la produttività delle applicazioni. Durante questa sessione, esaminiamo le ElastiCache best practice, i modelli di progettazione e gli anti-pattern.

[DAT3 ElastiCache 06—Approfondimento su Amazon \(re:Invent 2016\)](#)

DAT317—Come IFTTT utilizza ElastiCache (Redis) per prevedere gli eventi (re:Invent OSS 2016)

IFTTT è un servizio gratuito che consente alle persone di fare di più con i servizi che preferiscono, dall'automazione di attività semplici alla trasformazione del modo in cui qualcuno interagisce e controlla la propria casa. IFTTT utilizza ElastiCache (RedisOSS) per archiviare la cronologia delle esecuzioni delle transazioni e pianificare le previsioni, nonché gli indici per i documenti di registro su Amazon S3. Guarda questa sessione per scoprire come la potenza di scripting di Lua e i tipi di dati di Redis OSS hanno consentito alle persone di realizzare cose che non sarebbero state in grado di fare altrove.

[DAT317—Come IFTTT utilizza ElastiCache \(RedisOSS\) per prevedere gli eventi \(re:Invent 2016\)](#)

DAT407—Amazon ElastiCache Deep Dive (re:Invent 2015)

Dai un'occhiata dietro le quinte per scoprire il design e l'architettura ElastiCache di Amazon. Scopri i modelli di progettazione comuni delle nostre OSS offerte Memcached e Redis e come i clienti li hanno utilizzati per le operazioni in memoria e ottenuto latenza e throughput migliorati per le applicazioni. Durante questa sessione, esaminiamo le best practice, i modelli di progettazione e gli anti-pattern relativi ad Amazon ElastiCache.

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

SDD402—Amazon ElastiCache Deep Dive (re:Invent 2014)

In questo video, esaminiamo i casi d'uso più comuni della memorizzazione nella cache, i OSS motori Memcached e Redis, i modelli che ti aiutano a determinare quale motore è migliore per le tue esigenze, l'hashing coerente e altro ancora come mezzo per creare applicazioni veloci e

scalabili. Frank Wiebe, Principal Scientist di Adobe, spiega in dettaglio come Adobe utilizza Amazon ElastiCache per migliorare l'esperienza dei clienti e far crescere la propria attività.

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

DAT307 — Approfondimento ElastiCache sull'architettura e sui modelli di progettazione di Amazon (re:Invent 2013)

In questo video esamineremo caching, strategie di caching, dimensionamento e monitoraggio. Confrontiamo anche i motori Memcached e Redis. OSS Durante questa sessione, esaminiamo anche le migliori pratiche e i modelli di progettazione relativi ad Amazon ElastiCache.

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

Gestione dei nodi in ElastiCache

Un nodo è l'elemento costitutivo più piccolo di una ElastiCache distribuzione Amazon. È una porzione di rete sicura di dimensioni fisse. RAM Ogni nodo esegue il motore che è stato scelto al momento della creazione e dell'ultima modifica del cluster. Ogni nodo ha il proprio nome e porta Domain Name Service (). DNS Sono supportati diversi tipi di ElastiCache nodi, ciascuno con quantità variabili di memoria e potenza di calcolo associate.

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

In generale, grazie al supporto per lo sharding, le implementazioni Valkey o Redis OSS (abilitata la modalità cluster) dispongono di una serie di nodi più piccoli. Al contrario, le implementazioni Valkey o Redis OSS (modalità cluster disabilitata) hanno un numero inferiore di nodi più grandi in un cluster. Per una discussione dettagliata sulle dimensioni del nodo da utilizzare, consultare [Scelta delle dimensioni dei nodi](#).

Argomenti

- [Visualizzazione dello stato ElastiCache del nodo](#)
- [OSSNodi e shard Valkey o Redis](#)
- [Connessione ai nodi](#)
- [Tipi di nodi supportati](#)
- [Riavvio dei nodi](#)

- [Sostituzione dei nodi \(Valkey e RedisOSS\)](#)
- [Sostituzione dei nodi \(Memcached\)](#)
- [Nodi riservati](#)
- [Migrazione dei nodi della generazione precedente](#)

Alcune importanti operazioni che coinvolgono i nodi sono le seguenti:

- [Aggiungere nodi a un ElastiCache cluster](#)
- [Rimozione di nodi da un ElastiCache cluster](#)
- [Ridimensionamento ElastiCache](#)
- [Ricerca degli endpoint di connessione in ElastiCache](#)
- [Identifica automaticamente i nodi del cluster \(Memcached\)](#)

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 l'APIoperazione.

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.

Tipo	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.

Tipo	Fatturato	Descrizione
modifying	Fatturato	Il ElastiCache nodo viene modificato a causa di una richiesta del cliente di modificare il nodo.

Tipo	Fatturato	Descrizione
updating	Fatturato	<p>Uno stato di aggiornamento indica che una o più delle seguenti condizioni sono vere per il ElastiCache nodo Amazon:</p> <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 VPC di sicurezza 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 OSS password ElastiCache with Valkey o Redis viene aggiornata/ruotata utilizzando. AWS Secrets Manager

Tipo	Fatturato	Descrizione
<code>rebooting cache cluster nodes</code>	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.
<code>incompatible_parameters</code>	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.

Tipo	Fatturato	Descrizione
incompatible_network	Non fatturata	<p data-bbox="1068 226 1500 449">Uno stato di rete incompatibile indica che una o più delle seguenti condizioni sono vere per il nodo Amazon: ElastiCache</p> <ul data-bbox="1068 499 1468 919" style="list-style-type: none"><li data-bbox="1068 499 1451 674">• Non ci sono indirizzi IP disponibili nella sottorete in cui è stato avviato il ElastiCache nodo.<li data-bbox="1068 701 1468 919">• La sottorete menzionata nel gruppo di ElastiCache sottoreti non esiste più in Amazon Virtual Private Cloud (AmazonVPC).

Tipo	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.• Impossibile ripristinare l'RDB istantanea specificata.• L' AWS account per il ElastiCache cluster è stato sospeso.

Tipo	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 Valkey o RedisOSS.

Visualizzazione dello stato ElastiCache del nodo con la console

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

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Redis OSS 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
```

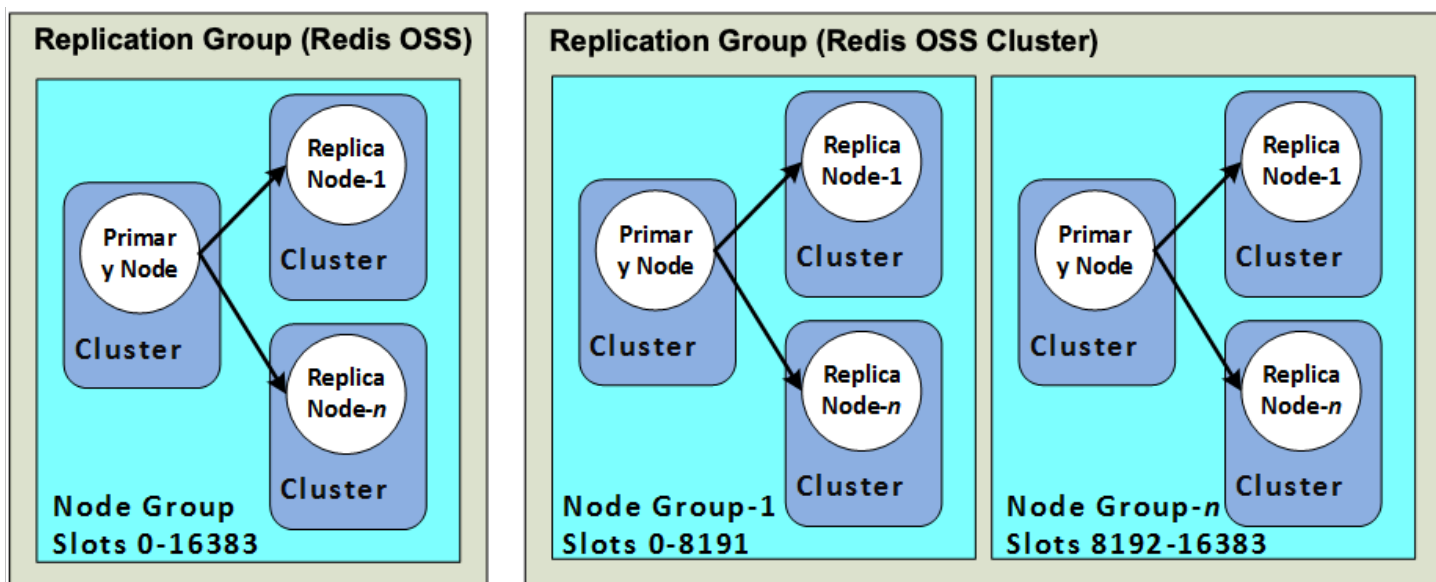
Visualizzazione dello stato del ElastiCache nodo tramite API

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

OSSNodi e shard Valkey o Redis

Uno shard (o meglio CLI, un gruppo di nodi) è una disposizione gerarchica di nodi, ciascuno racchiuso in un cluster. API 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. Valkey, o Redis OSS versione 3.2 e successive, supporta più shard all'interno di un cluster (nell'ambito di un gruppo di replica). API CLI Questo supporto consente il partizionamento dei dati in un cluster Valkey o Redis OSS (modalità cluster abilitata).

Il diagramma seguente illustra le differenze tra un cluster Valkey o Redis OSS (modalità cluster disabilitata) e un cluster Valkey o Redis (modalità cluster abilitata). OSS



I cluster Valkey o Redis OSS (modalità cluster abilitata) supportano la replica tramite shard.

L'API operazione [DescribeReplicationGroups](#) (CLI: [describe-replication-groups](#)) elenca i gruppi di nodi con i nodi membri, il ruolo del nodo all'interno del gruppo di nodi e anche altre informazioni.

Quando si crea un OSS cluster Valkey o Redis, si specifica se si desidera creare un cluster con il clustering abilitato. I cluster Valkey o Redis OSS (modalità cluster disabilitata) non hanno mai più di uno shard, che può essere scalato 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](#), [Aggiungere una replica di lettura per Valkey o Redis OSS \(modalità cluster disabilitata\)](#) o [Eliminazione di una replica di lettura per Valkey o Redis OSS \(modalità cluster disabilitata\)](#). I cluster Valkey o Redis OSS (modalità cluster disabilitata) possono anche scalare verticalmente modificando i tipi di nodi. Per ulteriori informazioni, consulta [Scalabilità dei nodi di replica per Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

Il limite di nodi o shard può essere aumentato fino a un massimo di 500 per cluster se il motore è Valkey o Redis versione 5.0.6 o successiva. OSS 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 insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo troppo piccolo o che le sottoreti sono condivise e utilizzate intensamente 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.

Dopo aver creato un cluster Valkey o Redis OSS (modalità cluster abilitata), è possibile modificarlo (ampliarlo o ridurlo). Per ulteriori informazioni, consulta [Ridimensionamento ElastiCache](#) e [Sostituzione dei nodi \(Valkey e RedisOSS\)](#).

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

Connessione ai nodi Valkey o Redis OSS

Prima di tentare di connettersi ai OSS nodi Valkey o Redis del cluster, è necessario disporre degli endpoint per i nodi. Per individuare gli endpoint, consultare la seguente documentazione:

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

Nell'esempio seguente, si utilizza l'utilità `valkey-cli` per connettersi a un cluster che esegue Valkey o Redis. OSS

Note

[Per ulteriori informazioni sui comandi disponibili, consultate la pagina Web Comandi.](#)

Per connettersi a un OSS cluster Valkey o Redis utilizzando `valkey-cli`

1. Connettiti alla tua EC2 istanza Amazon utilizzando l'utilità di connessione che preferisci.

Note

Per istruzioni su come connettersi a un'EC2 istanza Amazon, consulta la [Amazon EC2 Getting Started Guide](#).

2. Per creare `valkey-cli`, scaricare e installare la GNU Compiler Collection (`gcc`). Al prompt dei comandi dell'EC2 istanza, immettete il comando seguente e immettetelo `y` al prompt di conferma.

```
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. Scaricate e compilate l'utilità `valkey-cli`. Questa utilità è inclusa nella distribuzione del software Valkey. Al prompt dei comandi dell'EC2istanza, digita i seguenti comandi:

Note

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

```
wget https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean      # ubuntu systems only
make
```

4. Al prompt dei comandi dell'EC2istanza, digita il comando seguente.

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

Viene visualizzato un prompt dei comandi Valkey o Redis simile al seguente.

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


5. Verifica la connessione eseguendo i comandi Valkey o Redis. OSS

Ora sei connesso al cluster e puoi eseguire comandi Valkey o Redis. OSS Di seguito sono riportati alcuni comandi di esempio con le relative risposte Valkey o Redis. OSS

```
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 valkey-cli
```

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

Connessione ai nodi Memcached

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

- [Ricerca degli endpoint di un cluster \(console\) \(Memcached\)](#)
- [Individuazione degli endpoint \(AWS CLI\)](#)
- [Ricerca degli endpoint \(\) ElastiCache API](#)

L'esempio seguente utilizza l'utilità telnet per eseguire la connessione a un nodo che esegue Memcached.

Note

Per ulteriori informazioni su Memcached e i comandi Memcached disponibili, consulta il sito Web [Memcached](#).

Per connettersi a un nodo mediante telnet

1. Connettiti alla tua EC2 istanza Amazon utilizzando l'utilità di connessione che preferisci.

Note

Per istruzioni su come connettersi a un'EC2istanza Amazon, consulta la [Amazon EC2 Getting Started Guide](#).

2. Scarica e installa l'utilità telnet sulla tua EC2 istanza Amazon. Al prompt dei comandi dell'EC2istanza Amazon, digita il comando seguente e digita y al prompt dei comandi.

```
sudo yum install telnet
```

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: 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!
```

3. Al prompt dei comandi dell'EC2istanza Amazon, digita il comando seguente, sostituendo l'endpoint del nodo con quello mostrato in questo esempio.

```
telnet mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com 11211
```

Viene visualizzato un output simile al seguente.

```
Trying 128.0.0.1...
Connected to mycachecluster.eaogs8.0001.usw2.cache.amazonaws.com.
Escape character is '^]'.
>
```

4. Eseguire il test della connessione eseguendo i comandi Memcached.

Una volta connessi a un nodo è possibile eseguire i comandi Memcached. Di seguito è riportato un esempio.

```
set a 0 0 5      // Set key "a" with no expiration and 5 byte value
hello           // Set value as "hello"
STORED
get a           // Get value for key "a"
VALUE a 0 5
hello
END
get b           // Get value for key "b" results in miss
END
>
```

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 delle prestazioni per ogni tipo di nodo, consulta [Amazon EC2 Instance Types](#).

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

Argomenti

- [Generazione attuale \(Memcached\)](#)
- [Generazione attuale \(Valkey o Redis\) OSS](#)
- [Tipi di nodi supportati in base alla regione AWS](#)
- [Istanze a prestazioni espandibili](#)
- [Informazioni correlate](#)

Generazione attuale (Memcached)

Le tabelle seguenti mostrano la larghezza di banda di base e burst per i tipi di istanza che utilizzano il meccanismo di credito I/O di rete per superare la larghezza di banda di base.

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 Memcached	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.m7g.large		0,937	12,5
cache.m7g.xlarge		1,876	12,5

Tipo di istanza	Versione minima supportata di Memcached	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.m7g.2xlarge		3,75	15
cache.m7g.4xlarge		7,5	15
cache.m7g.8xlarge		15	N/D
cache.m7g.12xlarge		22,5	N/D
cache.m7g.16xlarge		30	N/D
cache.m6g.large	1,5,16	0.75	10,0
cache.m6g.xlarge	1,5,16	1,25	10,0
cache.m6g.2xlarge	1,5,16	2.5	10,0
cache.m6g.4xlarge	1,5,16	5.0	10,0
cache.m6g.8xlarge	1,5,16	12	N/D
cache.m6g.12xlarge	1,5,16	20	N/D
cache.m6g.16xlarge	1,5,16	25	N/D
cache.m5.large	1,5,16	0.75	10,0
cache.m5.xlarge	1,5,16	1,25	10,0
cache.m5.2xlarge	1,5,16	2.5	10,0
cache.m5.4xlarge	1,5,16	5.0	10,0
cache.m5.12xlarge	1,5,16	N/D	N/D
cache.m5.24xlarge	1,5,16	N/D	N/D
cache.m4.large	1,5,16	0,45	1.2

Tipo di istanza	Versione minima supportata di Memcached	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.m4.xlarge	1,5,16	0.75	2.8
cache.m4.2xlarge	1,5,16	1	10,0
cache.m4.4xlarge	1,5,16	2.0	10,0
cache.m4.10xlarge	1,5,16	5.0	10,0
cache.t4g.micro	1,5,16	0,064	5.0
cache.t4g.small	1,5,16	0,128	5.0
cache.t4g.medium	1,5,16	0,256	5.0
cache.t3.micro	1,5,16	0,064	5.0
cache.t3.small	1,5,16	0,128	5.0
cache.t3.medium	1,5,16	0,256	5.0
cache.t2.micro	1,5,16	0,064	1,024
cache.t2.small	1,5,16	0,128	1,024
cache.t2.medium	1,5,16	0,256	1,024

Memoria ottimizzata per Memcached

Tipo di istanza	Versione minima supportata	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r7g.large		0,937	12,5
cache.r7g.xlarge		1,876	12,5
cache.r7g.2xlarge		3,75	15

Tipo di istanza	Versione minima supportata	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r7g.4xlarge		7,5	15
cache.r7g.8xlarge		15	N/D
cache.r7g.12xlarge		22,5	N/D
cache.r7g.16xlarge		30	N/D
cache.r6g.large	1,5,16	0.75	10,0
cache.r6g.xlarge	1,5,16	1,25	10,0
cache.r6g.2xlarge	1,5,16	2.5	10,0
cache.r6g.4xlarge	1,5,16	5.0	10,0
cache.r6g.8xlarge	1,5,16	12	N/D
cache.r6g.12xlarge	1,5,16	20	N/D
cache.r6g.16xlarge	1,5,16	25	N/D
cache.r5.large	1,5,16	0.75	10,0
cache.r5.xlarge	1,5,16	1,25	10,0
cache.r5.2xlarge	1,5,16	2.5	10,0
cache.r5.4xlarge	1,5,16	5.0	10,0
cache.r5.12xlarge	1,5,16	20	N/D
cache.r5.24xlarge	1,5,16	25	N/D
cache.r4.large	1,5,16	0.75	10,0
cache.r4.xlarge	1,5,16	1,25	10,0
cache.r4.2xlarge	1,5,16	2.5	10,0

Tipo di istanza	Versione minima supportata	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r4.4xlarge	1,5,16	5.0	10,0
cache.r4.8xlarge	1,5,16	12	N/D
cache.r4.16xlarge	1,5,16	25	N/D

Rete ottimizzata per Memcached

Tipo di istanza	Versione minima supportata	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.c7gn.large	1.6.6	6,25	30
cache.c7gn.xlarge	1.6.6	12,5	40
cache.c7gn.2xlarge	1.6.6	25	50
cache.c7gn.4xlarge	1.6.6	50	N/D
cache.c7gn.8xlarge	1.6.6	100	N/D
cache.c7gn.12xlarge	1.6.6	150	N/D
cache.c7gn.16xlarge	1.6.6	200	N/D

Generazione attuale (Valkey o Redis) OSS

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 Redis minima supportata OSS	I/O avanzato (Redis OSS 5.0.6+)	TLS scaricamento (Redis 6.2.5+) OSS	Multiplexing I/O migliorato (Redis 7.0.4+) OSS	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
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

Tipo di istanza	Versione Redis minima supportata OSS	I/O avanzato (Redis OSS 5.0.6+)	TLSScaricamento (Redis 6.2.5+) OSS	Multiple larghezza di banda di I/O migliorato (Redis 7.0.4+) OSS	Larghezza di banda base (Gb/s)	Larghezza di banda burst (Gb/s)
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
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

Tipo di istanza	Versione Redis minima supportata OSS	I/O avanzato (Redis OSS 5.0.6+)	TLSScaricamento (Redis 6.2.5+) OSS	Multiple I/O migliorato (Redis 7.0.4+) OSS	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.m4.2xlarge	3.2.4	Y	Y	Y	1	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
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 Redis OSS minima supportata	I/O avanzato (Redis OSS 5.0.6+)	TLS scaricamento (Redis 6.2.5+) OSS	Multiplexing I/O migliorato (Redis 7.0.4+) OSS	Larghezza di banda di base (Gb/s)	Larghezza di banda di 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
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

Tipo di istanza	Versione Redis OSS minima supportata	I/O avanzato (Redis OSS 5.0.6+)	TLSScaricamento (Redis 6.2.5+) OSS	Multiple I/O migliorato (Redis 7.0.4+) OSS	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
cache.r6g.8xlarge	5.0.6	Y	Y	Y	12	N/D
cache.r6g.12xlarge	5.0.6	Y	Y	Y	20	N/D
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

Tipo di istanza	Versione Redis OSS minima supportata	I/O avanzato (Redis OSS 5.0.6+)	TLS scaricamento (Redis 6.2.5+) OSS	Multiplexing I/O migliorato (Redis 7.0.4+) OSS	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
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 Redis OSS minima supportata	I/O avanzato (Redis OSS 5.0.6+)	TLS scaricamento (Redis 6.2.5+) OSS	Multiplexing I/O migliorato (Redis 7.0.4+) OSS	Larghezza di banda di 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

Tipo di istanza	Versione Redis OSS minima supportata	I/O avanzato (Redis OSS 5.0.6+)	TLSScaricamento (Redis 6.2.5+) OSS	Multiple I/O miglioramento (Redis 7.0.4+) OSS	Larghezza di banda di base (Gb/s)	Larghezza di banda burst (Gb/s)
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 Redis OSS minima supportata	I/O avanzato (Redis OSS 5.0.6+)	TLSScaricamento (Redis 6.2.5+) OSS	Multiple I/O miglioramento (Redis 7.0.4+) OSS	Larghezza di banda di base (Gb/s)	Larghezza di banda 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

Tipo di istanza	Versione Redis OSS minima supportata	I/O avanzato (Redis OSS 5.0.6+)	TLS scaricamento (Redis 6.2.5+) OSS	Multiplexing I/O migliorato (Redis 7.0.4+) OSS	Larghezza di banda di base (Gb/s)	Larghezza di banda di burst (Gb/s)
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 tra AWS le regioni. Per maggiori dettagli, consulta i [ElastiCacheprezzi di Amazon](#).

Istanze a prestazioni espandibili

Puoi avviare nodi di cache T4g, T3-Standard e T2-Standard espandibili per uso generico in Amazon. ElastiCache Questi nodi forniscono un livello di CPU prestazioni di base con la possibilità di interrompere l'CPUutilizzo in qualsiasi momento, fino all'esaurimento dei crediti accumulati. Un CPUcredito fornisce le prestazioni di un core completo CPU per un minuto.

I nodi T4g, T3 e T2 di Amazon ElastiCache sono configurati come standard e adatti a carichi di lavoro con un CPU utilizzo medio costantemente inferiore alle prestazioni di base dell'istanza. Per superare la linea di base, il nodo spende i crediti accumulati nel suo saldo di crediti. 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 subisca un brusco calo delle prestazioni quando il saldo del credito accumulato si esaurisce. CPU Per ulteriori informazioni, consulta [CPUcrediti e prestazioni di base per istanze Burstable Performance](#) nella Amazon EC2 User Guide.

La tabella seguente elenca i tipi di nodi Burstable Performance e la velocità con cui vengono guadagnati CPU crediti all'ora. Mostra anche il numero massimo di CPU crediti guadagnati che un nodo può accumulare e il numero di crediti vCPUs per nodo. Inoltre, fornisce il livello di prestazioni di base come percentuale delle prestazioni di base complete (utilizzando una sola vCPU).

CPU crediti guadagnati all'ora	Quantità massima di crediti guadagnati che può essere accumulati*	vCPUs	Prestazioni di base per v CPU	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
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

CPUcrediti guadagnati all'ora	Quantità massima di crediti guadagnati che può essere accumulata*	vCPUs	Prestazioni di base per v CPU	Memoria (GiB)	Prestazioni di rete
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 sono per v. CPU Alcune dimensioni dei nodi che hanno più di una v. CPU Per questi, calcola l'CPUutilizzo di base del nodo moltiplicando la CPU percentuale v per il numero di vCPUs

Le seguenti metriche CPU di credito sono disponibili per le istanze a prestazioni espandibili T3 e T4g:

Note

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

- CPUcreditUsage
- CPUcreditBalance

[Per ulteriori informazioni su queste metriche, consulta Credit Metrics. CPU](#)

Inoltre, ricordare i seguenti dettagli:

- Per impostazione predefinita, tutti i tipi di nodi dell'attuale generazione vengono creati in un cloud privato virtuale (VPC) basato su AmazonVPC.

- I file Redis OSS append-only (AOF) non sono supportati per le istanze T2. Le variabili di OSS configurazione Redis `appendonly` e `appendfsync` non sono supportate nella versione Redis 2.8.22 e successive. OSS

Informazioni correlate

- [Caratteristiche e dettagli ElastiCache del prodotto Amazon](#)
- [Parametri specifici del tipo di nodo Memcached per Memcached](#)
- [Parametri Valkey e Redis OSS](#)
- [Crittografia in TLS transito \(\)](#)

Riavvio dei nodi

Alcune modifiche richiedono il riavvio di un cluster Redis OSS o Memcached per poter applicare le modifiche. Ad esempio, per alcuni parametri la modifica del valore in un gruppo di parametri si applica solo dopo un riavvio.

Argomenti

- [Riavvio dei nodi OSS Redis \(solo modalità cluster disabilitata\)](#)
- [Riavvio di un cluster per Memcached](#)

Riavvio dei nodi OSS Redis (solo modalità cluster disabilitata)

Per i cluster Valkey o Redis OSS (modalità cluster disabilitata), i parametri nei gruppi di parametri che vengono applicati solo dopo il riavvio sono:

- `activeresharding`
- `database`

I nodi Redis possono essere aggiornati solo tramite la console. ElastiCache È possibile riavviare un solo nodo alla volta. Per riavviare più nodi, è necessario ripetere la procedura per ogni nodo.

Modifiche ai parametri Valkey o Redis OSS (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri su un cluster Valkey o Redis OSS (modalità cluster abilitata), segui i passaggi seguenti.

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

Le modifiche ad altri parametri non richiedono questo.

Utilizzando il AWS Management Console

È possibile riavviare un nodo utilizzando la ElastiCache console.

Per riavviare un nodo (console)

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione desiderata.
3. Nel riquadro di navigazione a sinistra, scegli Redis. OSS

Viene visualizzato un elenco di cluster che eseguono RedisOSS.

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.

Riavvio di un cluster per Memcached

Quando si riavvia un cluster Memcached, il cluster cancella tutti i dati e riavvia il motore. Nel corso di questo processo non è possibile accedere al cluster. Dato che il cluster elimina tutti i dati, quando è nuovamente disponibile è vuoto.

È possibile riavviare un cluster utilizzando la console, il, o il ElastiCache . AWS CLI ElastiCache API. Sia che si utilizzi la ElastiCache console, la AWS CLI o la ElastiCache API, è possibile avviare il riavvio di un solo cluster. Per riavviare più cluster si deve ripetere il processo o l'operazione.

Utilizzando il AWS Management Console

È possibile riavviare un cluster utilizzando la ElastiCache console.

Per riavviare un cluster (console)

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione che ti interessa.
3. Nel pannello di navigazione, scegliere il motore in esecuzione sul cluster da riavviare.

Viene visualizzato un elenco dei cluster che eseguono il motore scelto.

4. Scegliere il cluster da riavviare selezionando la casella a sinistra del nome del cluster.

Il pulsante Reboot (Riavvia) diventa attivo.

Se si sceglie più di un cluster, il pulsante Reboot (Riavvia) non è attivo.

5. Scegliere Reboot (Riavvia).

Viene visualizzata una schermata di conferma del riavvio del cluster.

6. Per riavviare il cluster, scegliere Reboot (Riavvia). Lo stato del cluster cambia in fase di riavvio dei nodi del cluster.

Per non riavviare il cluster, scegliere Cancel (Annulla).

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

Per riavviare un nodo specifico, seleziona il nodo e scegliere Riavvio.

Usando il AWS CLI

Per riavviare un cluster (AWS CLI), utilizzare l'operazione `reboot-cache-cluster`.

Per riavviare nodi specifici nel cluster, utilizzare l'operazione `--cache-node-ids-to-reboot` per elencare cluster specifici da riavviare. Il comando seguente riavvia i nodi 0001, 0002 e 0004 del cluster `my-cluster`.

Per Linux, macOS o Unix:

```
aws elasticache reboot-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --cache-node-ids-to-reboot 0001 0002 0004
```

Per Windows:

```
aws elasticache reboot-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --cache-node-ids-to-reboot 0001 0002 0004
```

Per riavviare tutti i nodi del cluster, utilizzare il parametro `--cache-node-ids-to-reboot` ed elencare tutti gli id dei nodi del cluster. Per ulteriori informazioni, vedere [reboot-cache-cluster](#).

Utilizzando il ElastiCache API

Per riavviare un cluster utilizzando ElastiCache API, utilizzare l'operazione `RebootCacheCluster`.

Per riavviare nodi specifici nel cluster, utilizzare l'operazione `CacheNodeIdsToReboot` per elencare cluster specifici da riavviare. Il comando seguente riavvia i nodi 0001, 0002 e 0004 del cluster `my-cluster`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RebootCacheCluster  
&CacheClusterId=my-cluster  
&CacheNodeIdsToReboot.member.1=0001  
&CacheNodeIdsToReboot.member.2=0002  
&CacheNodeIdsToReboot.member.3=0004  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z
```

```
&X-Amz-Credential=<credential>
```

Per riavviare tutti i nodi del cluster, utilizzare il parametro `CacheNodeIdsToReboot` ed elencare tutti gli id dei nodi del cluster. Per ulteriori informazioni, vedere [RebootCacheCluster](#).

Sostituzione dei nodi (Valkey e RedisOSS)

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

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

Note

I nodi di cache sostitutivi generati automaticamente da Amazon ElastiCache possono 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.

L'elenco seguente identifica le azioni che puoi intraprendere quando ElastiCache pianifica la sostituzione di uno dei tuoi OSS nodi Valkey o Redis. Per velocizzare l'individuazione delle informazioni richieste per questa situazione, scegliere dal seguente menu.

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

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

Opzioni di sostituzione dei nodi Valkey e Redis OSS

- Non eseguire alcuna operazione: se non si esegue alcuna operazione, ElastiCache sostituisce il nodo come pianificato.

Per le configurazioni non cluster con l'autofailover abilitato, i cluster su Valkey 7.2 e versioni successive e Redis OSS 5.0.6 e versioni successive completano la sostituzione mentre il cluster continua a rimanere online e a soddisfare le richieste di scrittura in entrata. Per i cluster abilitati al failover automatico su Redis OSS 4.0.10 o versioni precedenti, potresti notare una breve interruzione di scrittura, fino a pochi secondi, associata agli aggiornamenti. DNS

Se il nodo è membro di un cluster abilitato al failover automatico, ElastiCache con Valkey o Redis OSS offre una maggiore disponibilità durante l'applicazione di patch, aggiornamenti e altre sostituzioni dei nodi relative alla manutenzione.

Per le configurazioni Valkey o Redis OSS Cluster configurate per l'uso ElastiCache ElastiCache con i client Valkey o Redis Cluster, la sostituzione ora viene completata mentre il OSS cluster soddisfa le richieste di scrittura in entrata.

Per le configurazioni non cluster con autofailover abilitato, i cluster su Valkey 7.2 e versioni successive e Redis OSS 5.0.6 e versioni successive vengono sostituiti completamente mentre il cluster continua a rimanere online e a soddisfare le richieste di scrittura in entrata. Per i cluster abilitati al failover automatico su Redis OSS 4.0.10 o versioni precedenti, potresti notare una breve interruzione di scrittura, fino a pochi secondi, associata agli aggiornamenti. DNS

Se il nodo è autonomo, Amazon avvia ElastiCache prima un nodo sostitutivo e poi si sincronizza dal nodo esistente. Durante tale periodo il nodo esistente non sarà disponibile per richieste di servizio. Una volta completata la sincronizzazione, il nodo esistente viene terminato e il nuovo nodo prende il suo posto. ElastiCache fa del suo meglio per conservare i dati durante questa operazione.

- **Modifica della finestra di manutenzione:** per gli eventi di manutenzione programmata, si riceve 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 ElastiCache notifica 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, consulta [Gestione della manutenzione dei ElastiCache cluster](#).

- Sostituisci l'unico nodo in qualsiasi OSS cluster Valkey o Redis: se il cluster non dispone di repliche di lettura, puoi 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, consulta [Esecuzione di backup manuali](#).
 2. Creare un nuovo cluster inizializzandolo dalla snapshot. Per istruzioni, consulta [Ripristino da un backup in una nuova cache](#).
 3. Eliminare il cluster con il nodo pianificato per la sostituzione. Per istruzioni, consulta [Eliminazione di un cluster in ElastiCache](#).
 4. Nell'applicazione, sostituire l'endpoint del vecchio nodo con l'endpoint del nuovo nodo.
- Sostituisci un nodo di replica in qualsiasi cluster Valkey o Redis: per sostituire un OSS cluster di replica, aumenta 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 Valkey o Redis OSS

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.
- Sostituisci qualsiasi nodo in uno shard Valkey o Redis OSS (modalità cluster abilitata): per sostituire il nodo in un cluster senza tempi di inattività, utilizza il resharding 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 Valkey o Redis (modalità cluster abilitata) OSS

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.
- Sostituisci un nodo in un cluster Valkey o Redis OSS (modalità cluster disabilitata): se il cluster è un cluster Valkey o Redis OSS (modalità cluster disabilitata) senza repliche di lettura, utilizza la seguente procedura per sostituire il nodo.

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, consulta [Per aggiungere la replica a un cluster Valkey o Redis senza shard OSS](#).
 2. Aggiungere una replica di lettura al cluster. Per istruzioni, consulta [Per aggiungere nodi a un ElastiCache cluster \(console\)](#).
 3. Promuovere la nuova replica di lettura creata al nodo primario. Per istruzioni, consulta [Promozione di una replica di lettura a quella primaria, per i gruppi di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#).
 4. Eliminare il nodo pianificato per la sostituzione. Per istruzioni, consulta [Rimozione di nodi da un ElastiCache cluster](#).
 5. Nell'applicazione, sostituire l'endpoint del vecchio nodo con l'endpoint del nuovo nodo.
- Sostituisci una replica di lettura Valkey o Redis OSS (modalità cluster disabilitata): se il nodo è una replica in lettura, sostituisci 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, consulta [Modifica di un gruppo di replica](#).

Per sostituire una replica Valkey o Redis (modalità cluster disabilitata) leggi la replica OSS

1. Eliminare la replica che è stata pianificata per la sostituzione. Per le istruzioni, consulta quanto segue:
 - [Riduzione del numero di repliche in uno shard](#)
 - [Rimozione di nodi da un ElastiCache 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, consulta quanto segue:
 - [Incremento del numero di repliche in uno shard](#)
 - [Aggiungere una replica di lettura per Valkey o Redis OSS \(modalità cluster disabilitata\)](#)
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, consulta [Abilitazione della funzione Multi-AZ](#) .
- Sostituisci un nodo primario Valkey o Redis OSS (modalità cluster disabilitata): se il nodo è il nodo principale, promuovi innanzitutto una replica di lettura a principale. 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, consulta [Modifica di un gruppo di replica](#).

Per sostituire un nodo primario Valkey o Redis OSS (modalità cluster disabilitata)

1. Promuovere una replica di lettura al nodo primario. Per istruzioni, consulta [Promozione di una replica di lettura a quella primaria, per i gruppi di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#).
2. Eliminare il nodo pianificato per la sostituzione (il vecchio nodo primario). Per istruzioni, consulta [Rimozione di nodi da un ElastiCache 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, consulta [Aggiungere una replica di lettura per Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

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](#) .

Sostituzione dei nodi (Memcached)

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

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

Note

I nodi di cache sostitutivi generati automaticamente da Amazon ElastiCache possono 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.

L'elenco seguente identifica le azioni che è possibile intraprendere quando si ElastiCache pianifica la sostituzione di uno dei nodi Memcached.

- **Non fare nulla:** se non fai nulla, ElastiCache sostituisce il nodo come pianificato. Quando sostituisce ElastiCache automaticamente il nodo con un nuovo nodo, il nuovo nodo è inizialmente vuoto.
- **Modifica della finestra di manutenzione:** per gli eventi di manutenzione programmata, ricevi un'e-mail o un evento di notifica da ElastiCache. In questo caso, se si modifica la finestra di manutenzione prima dell'orario di sostituzione pianificato, il nodo viene sostituito al nuovo orario. Per ulteriori informazioni, consulta [Modifica di un cluster ElastiCache](#).

Note

La possibilità di modificare la finestra di sostituzione spostando la finestra di manutenzione è disponibile solo quando la ElastiCache notifica 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, consulta [Gestione della manutenzione dei ElastiCache cluster](#).

- Sostituire manualmente il nodo : Se è necessario sostituire il nodo prima della finestra di manutenzione successiva, sostituire manualmente il nodo.

Se sostituisci manualmente il nodo, le chiavi vengono ridistribuite. La ridistribuzione causa degli eventi di cache miss.

Per sostituire manualmente un nodo Memcached

1. Eliminare il nodo pianificato per la sostituzione. Per istruzioni, consulta [Rimozione di nodi da un ElastiCache cluster](#).
2. Aggiungere un nuovo nodo al cluster. Per istruzioni, consulta [Aggiungere nodi a un ElastiCache cluster](#).
3. Se non stai utilizzando l'individuazione automatica sul cluster, vedi l'applicazione e sostituisci ogni istanza dell'endpoint del nodo precedente con l'endpoint del nuovo nodo.

Nodi riservati

Prenotare uno o più ElastiCache nodi potrebbe 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 la pagina [ElastiCache dei prezzi di Amazon](#).

Argomenti

- [Gestione dei costi con nodi riservati](#)
- [Offerte di nodi riservati standard](#)
- [Dimensioni dei nodi riservati flessibili](#)
- [Eliminazione di un nodo riservato](#)
- [Offerte di nodi riservati legacy](#)
- [Recupero di informazioni sulle offerte di nodi riservati](#)
- [Acquisto di un nodo riservato](#)
- [Recupero di informazioni sui nodi riservati](#)

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 la pagina [ElastiCache dei prezzi di Amazon](#).

AWS La regione, il tipo di nodo e la durata del periodo devono essere scelti al momento dell'acquisto e non possono essere modificati in seguito.

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

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

Offerte di nodi riservati standard

Quando acquisti un'istanza di nodo riservato (RI) in Amazon ElastiCache, puoi sottoscrivere un impegno a ottenere una tariffa scontata su un tipo di istanza di nodo e una AWS regione specifici per la durata dell'istanza del nodo riservato. Per utilizzare un'istanza di nodo ElastiCache riservato Amazon, crei una nuova istanza di ElastiCache nodo, proprio come faresti per un'istanza on-demand.

Se le specifiche della nuova istanza del nodo di riserva corrispondono a un'istanza di nodo riservato esistente per il tuo account, ti verrà addebitata la tariffa scontata offerta per l'istanza del nodo riservato. In caso contrario, l'istanza di nodo viene fatturata in base a una tariffa on demand. Questi standard RIs sono disponibili a partire dalle famiglie di istanze R5 e M5.

Note

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

Tipi offerta

No Upfront RI fornisce l'accesso a un'ElastiCache istanza riservata senza richiedere un pagamento anticipato. L' ElastiCache istanza riservata No Upfront fattura una tariffa oraria scontata per ogni ora entro il termine, indipendentemente dall'utilizzo.

Partial Upfront RI richiede il pagamento anticipato di una parte dell' ElasticCache istanza 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.

Dimensioni dei nodi riservati flessibili

Tutti i nodi riservati sono flessibili in termini di dimensioni. Quando acquisti un nodo riservato, una cosa che specifichi è il tipo di nodo, ad esempio `cache.r6g.xlarge`. Per ulteriori informazioni sui tipi di nodi, consulta la pagina [ElastiCache dei prezzi di Amazon](#).

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

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

Aggiornamento OSS dei nodi riservati da Redis a Valkey

Con il lancio di Valkey nel ElastiCache, ora puoi applicare lo sconto Redis sul nodo OSS riservato al motore di cache Valkey. Puoi passare da Redis a Valkey pur continuando OSS a beneficiare dei contratti e delle prenotazioni esistenti. Oltre a poter sfruttare i vantaggi offerti dalla famiglia e dal motore di cache node, è possibile ottenere anche un valore incrementale maggiore. Valkey ha un prezzo scontato del 20% rispetto a Redis e OSS, grazie alla flessibilità dei nodi riservati, puoi utilizzare i tuoi nodi OSS riservati Redis per coprire il 20% in più di nodi Valkey in esecuzione.

Per calcolare la tariffa scontata, ogni combinazione di ElastiCache nodo e motore ha un fattore di normalizzazione misurato in unità. Le unità dei nodi riservati possono essere applicate a qualsiasi nodo in esecuzione all'interno della famiglia di istanze del nodo riservato per un determinato motore. I nodi OSS riservati Redis possono inoltre essere applicati a tutti i motori per coprire i nodi Valkey in esecuzione. Poiché Valkey ha un prezzo scontato rispetto a Redis OSS e Memcached, le sue unità per un determinato tipo di istanza sono inferiori, il che consente a un nodo OSS riservato Redis di coprire più nodi Valkey.

Ad esempio, supponiamo che tu abbia acquistato un nodo riservato per un nodo Redis `cache.r7g.4xlarge` per il OSS motore Redis (32 unità) e che tu stia utilizzando un OSS nodo Redis `cache.r7g.4xlarge` (32 unità). Se si aggiorna il nodo a Valkey, il fattore di normalizzazione del nodo

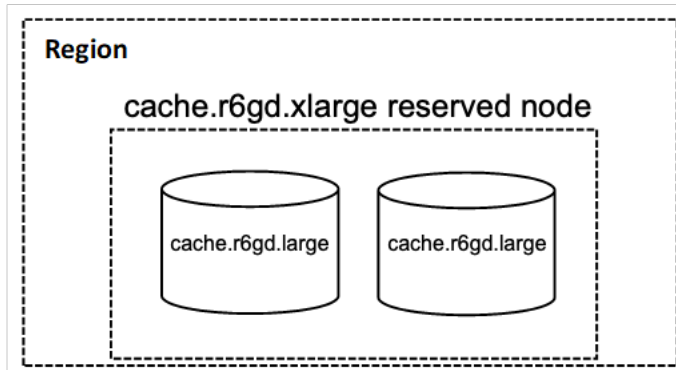
in esecuzione scende a 25,6 unità e il nodo riservato esistente fornisce altre 6,4 unità da utilizzare contro qualsiasi altro OSS nodo Valkey o Redis in esecuzione all'interno della famiglia cache.r7g nella regione. Puoi usarlo per coprire il 25% di un altro nodo Valkey cache.r7g.4xlarge nell'account (25,6 unità) o il 100% di un nodo Valkey cache.r7g.xlarge (6,4 unità).

Confronto dell'utilizzo con le unità normalizzate

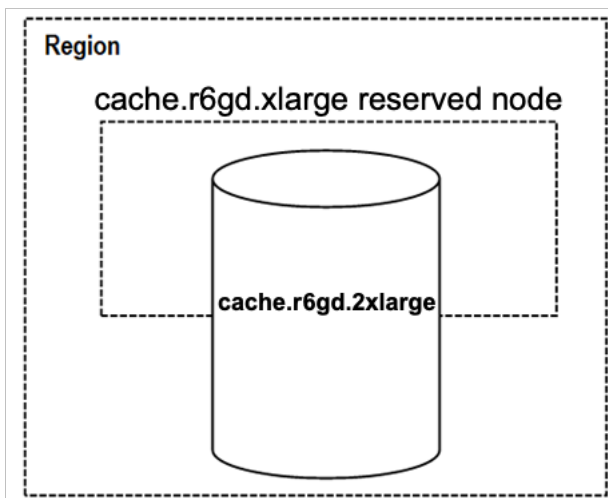
È possibile confrontare l'utilizzo per diverse dimensioni dei nodi riservati utilizzando unità normalizzate. Ad esempio, un'ora di utilizzo su due nodi cache.r6g.4xlarge equivale a 16 ore di utilizzo su un cache.r6g.large. La tabella seguente mostra il numero di unità normalizzate per ogni dimensione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

Dimensioni nodo	Unità normalizzate (RedisOSS, Memcached)	Unità normalizzate (Valkey)
micro	0,5	0.4
small	1	8.
medium	2	1.6
large	4	3.2
xlarge	8	6.4
2xlarge	16	12.8
4xlarge	32	25,6
6xlarge	48	38,4
8xlarge	64	51,2
10xlarge	80	64
12xlarge	96	76,8
16xlarge	128	102,4
24xlarge	192	153,6

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



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



Eliminazione di un nodo riservato

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

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

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 la pagina [ElastiCache dei prezzi di Amazon](#).

Recupero di informazioni sulle offerte di nodi riservati

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

Gli esempi seguenti mostrano come ottenere prezzi e informazioni sulle offerte disponibili di nodi riservati utilizzando AWS Management Console AWS CLI, e ElastiCache API.

Argomenti

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

Recupero di informazioni sulle offerte di nodi riservati

Per ottenere prezzi e altre informazioni sulle offerte di cluster riservati disponibili utilizzando il AWS Management Console, utilizzare la procedura seguente.

Per recuperare informazioni sulle offerte di nodi riservati disponibili

1. Accedi a AWS Management Console e apri la ElastiCache console 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. Per Engine, scegli Valkey, Redis o OSS Memcached.
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 prezzi e altre informazioni sulle offerte di nodi riservati disponibili per Valkey o RedisOSS, digita il seguente comando al prompt dei comandi:

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

Questa operazione produce 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 ottenere prezzi e altre informazioni sulle offerte di nodi riservati disponibili per Memcached, digitate il seguente comando al prompt dei comandi:

```

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

```

```

    ]
  },
  {
    "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
    "CacheNodeType": "cache.xx.12xlarge",
    "Duration": 31536000,
    "FixedPrice": X.X,
    "UsagePrice": X.X,
    "ProductDescription": "memcached",
    "OfferingType": "No Upfront",
    "RecurringCharges": [
      {
        "RecurringChargeAmount": X.XXXX,
        "RecurringChargeFrequency": "Hourly"
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta [describe-reserved-cache-nodes-offerings](#) nel Reference. AWS CLI

Ottenere informazioni sulle offerte dei nodi riservati () ElastiCache API

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, vedere [DescribeReservedCacheNodesOfferings](#) nel ElastiCache API Reference.

Acquisto di un nodo riservato

Gli esempi seguenti mostrano come acquistare un'offerta di nodi riservati utilizzando il AWS Management Console AWS CLI, il e il ElastiCache API.

Important

Gli esempi riportati in questa sezione comportano addebiti sul tuo AWS account che non puoi annullare.

Argomenti

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

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 procedura seguente utilizza l'offerta AWS Management Console per acquistare il nodo riservato offrendo id.

Per acquistare nodi riservati

1. Accedi a AWS Management Console e apri la ElastiCache console 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. Per Engine, scegli Valkey, Redis o OSS Memcached.
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, vedere [purchase-reserved-cache-nodes-offering](#) nel AWS CLI Reference.

Acquisto di un nodo riservato (ElastiCache API)

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, vedere [PurchaseReservedCacheNodesOffering](#) nel ElastiCache API Reference.

Recupero di informazioni sui nodi riservati

Puoi ottenere informazioni sui nodi riservati che hai acquistato utilizzando il AWS Management Console AWS CLI, il e il ElastiCache API.

Argomenti

- [Recupero di informazioni sui nodi riservati \(Console\)](#)
- [Recupero di informazioni sui nodi riservati \(AWS CLI\)](#)
- [Ottenere informazioni sui nodi riservati \(ElastiCache API\)](#)

Recupero di informazioni sui nodi riservati (Console)

La procedura seguente descrive come utilizzare AWS Management Console per ottenere informazioni sui nodi riservati acquistati.

Per ottenere informazioni sui nodi riservati che hai acquistato

1. Accedi a AWS Management Console e apri la ElastiCache console 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 il tuo AWS account, digita il seguente comando al prompt dei comandi:

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

Questa operazione produce un output simile al seguente (JSONformato):

```
{
  "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, vedere [describe-- reserved-cache-nodes](#) nel AWS CLI Reference.

Ottenere informazioni sui nodi riservati (ElastiCache API)

Per ottenere informazioni sui nodi riservati per il tuo AWS account, chiama l'DescribeReservedCacheNodesoperazione.

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 la [DescribeReservedCacheNodes](#)sezione ElastiCache API Reference.

Migrazione dei nodi della generazione precedente

I nodi di generazione precedente sono tipi di nodi che vengono eliminati gradualmente. Se non sono presenti cluster esistenti che utilizzano un tipo di nodo di generazione precedente, ElastiCache non supporta la creazione di nuovi cluster con quel 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 nodi supportati e dei tipi di nodi della generazione precedente in ElastiCache, consulta. [Tipi di nodi supportati](#)

Migrazione dei nodi su un cluster Valkey o Redis OSS

La procedura seguente descrive come migrare il tipo di nodo del OSS cluster Valkey o Redis utilizzando la Console. ElastiCache Durante questo processo, il OSS cluster Valkey o Redis continuerà a soddisfare 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 richiedere circa 60 secondi, principalmente a causa della propagazione. DNS
- La modalità cluster disabilitata (con nodo di replica) può richiedere circa 1 secondo per i cluster che eseguono Valkey 7.2 e versioni successive o Redis 5.0.6 e versioni successive. OSS Tutte le versioni precedenti possono durare circa 10 secondi.
- La modalità cluster abilitata può vedere circa 1 secondo.

Per modificare un tipo di nodo del OSS cluster Valkey o Redis utilizzando la console:

1. Accedi alla console e apri la ElastiCache console all'indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
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 Valkey o Redis utilizzando OSS: AWS CLI

Utilizzate [modify-replication-group](#) API 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 verso cui stai migrando. 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.

Correggere o rimuovere `restore-failed-node`

La procedura seguente descrive come correggere o rimuovere i nodi con errore di ripristino dal cluster Valkey o Redis. OSS Per ulteriori informazioni su come i ElastiCache nodi entrano in uno stato di ripristino fallito, consulta. [Visualizzazione dello stato ElastiCache del nodo](#) Ti consigliamo innanzitutto di rimuovere tutti i nodi in uno stato di ripristino fallito, quindi di migrare i nodi di generazione precedente rimanenti nel ElastiCache cluster a un tipo di nodo di nuova generazione e infine aggiungere nuovamente il numero richiesto di nodi.

Per rimuovere il nodo di ripristino non riuscito (console):

1. Accedi alla console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
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 ElastiCache cluster, ora puoi migrare a un tipo di nuova generazione. Per ulteriori informazioni, consulta la sezione precedente [Migrazione dei nodi su un cluster Valkey o Redis OSS](#).

Per aggiungere nuovamente nodi al cluster, consulta. ElastiCache [Aggiungere nodi a un ElastiCache cluster](#)

Migrazione dei nodi in un cluster Memcached

Per migrare ElastiCache (Memcached) a un tipo di nodo diverso, è necessario creare un nuovo cluster, che all'inizio è sempre vuoto e può essere popolato dall'applicazione.

Per migrare il tipo di nodo del cluster ElastiCache (Memcached) utilizzando la Console: ElastiCache

- Creare un nuovo cluster con il nuovo tipo di nodo. Per ulteriori informazioni, consulta [Creazione di un cluster Memcached \(console\)](#).
- Nell'applicazione, aggiornare gli endpoint con gli endpoint del nuovo cluster. Per ulteriori informazioni, consulta [Ricerca degli endpoint di un cluster \(console\) \(Memcached\)](#)
- Eliminare il cluster precedente. Per ulteriori informazioni, consulta [Eliminazione di un cluster in ElastiCache](#)

Gestione dei cluster in ElastiCache

Un cluster è una raccolta di uno o più nodi di cache, che eseguono tutti un'istanza del software del motore Valkey, Redis OSS o Memcached. Quando si crea un cluster, occorre specificare il motore e la versione che verrà utilizzata da tutti i nodi.

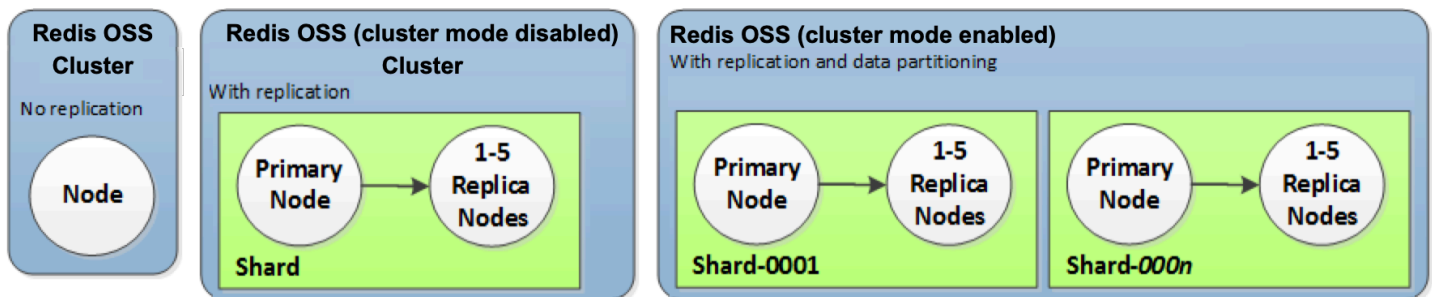
Cluster Valkey e Redis OSS

Il diagramma seguente illustra un tipico cluster Valkey o Redis. OSS Questi cluster possono contenere un singolo nodo o fino a sei nodi all'interno di uno shard (API/CLI: gruppo di nodi), un cluster Valkey o Redis a nodo singolo OSS (modalità cluster disabilitata) non ha shard e un cluster Valkey o Redis multinodo (modalità cluster disabilitata) ha un singolo shard. OSS I cluster Valkey o Redis OSS (modalità cluster abilitata) possono avere fino a 500 shard, con i dati partizionati tra gli shard. Il limite di nodi o shard può essere aumentato fino a un massimo di 500 per cluster se la versione del motore è Valkey 7.2 e successiva o Redis 5.0.6 e successiva. OSS 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 insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo 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.

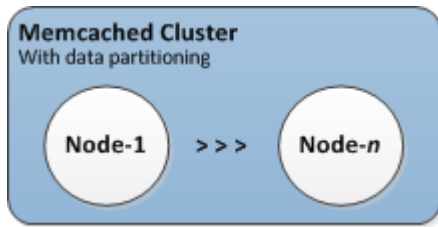
Quando sono presenti più nodi in uno OSS shard Valkey o Redis, uno dei nodi è un nodo primario di lettura/scrittura. Tutti gli altri nodi nella partizione sono repliche di sola lettura.

I cluster Valkey o Redis tipici hanno il seguente aspetto. OSS



Cluster Memcached

I cluster Memcached tipici hanno il seguente aspetto. I cluster Memcached contengono da 1 a 60 nodi, sui quali è possibile partizionare orizzontalmente i dati.



Operazioni Elasticache per Valkey, Redis e Memcached OSS

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 i ElastiCache API comandi and. 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 tramite un'EC2istanza Amazon. Se avvii il tuo cluster in un cloud privato virtuale (VPC) basato sul VPC servizio Amazon, puoi accedervi dall'esterno AWS. Per ulteriori informazioni, consulta [Accesso alle ElastiCache risorse dall'esterno AWS](#).

Per un elenco delle versioni supportate, consulta [Motori e versioni supportatiVersioni Redis supportate OSS](#), e [Versioni supportate ElastiCache \(Memcached\)](#).

Scelta del tipo di rete in ElastiCache

ElastiCache supporta le versioni 4 e 6 (IPv4eIPv6) del protocollo Internet, che consente di configurare il cluster in modo che accetti:

- solo IPv4 connessioni,
- solo IPv6 connessioni,
- entrambi IPv4 e IPv6 connessioni (dual-stack)

[IPv6 è supportato per carichi di lavoro che utilizzano Valkey 7.2 e versioni successive o il OSS motore Redis versione 6.2 e successive su tutte le istanze basate sul sistema Nitro.](#) Non sono previsti costi aggiuntivi per l'accesso a Over. ElastiCache IPv6

Note

La migrazione di 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.

[IPv6 è supportato per i carichi di lavoro che utilizzano il motore Memcached dalla versione 1.6.6 in poi su tutte le istanze basate sul sistema Nitro.](#) Non sono previsti costi aggiuntivi per l'accesso tramite. ElastiCache IPv6

Configurazione di sottoreti per tipo di rete

Se crei un cluster in AmazonVPC, devi specificare un gruppo di sottoreti. 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 entrambi IPv6 gli indirizzi assegnati per funzionare in modalità dual-stack IPv4 e una sottorete -only per funzionare solo come -only. IPv6 IPv6

Utilizzo di dual-stack

Quando si utilizza ElastiCache (RedisOSS) in modalità cluster abilitata, dal punto di vista di un'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.

Quando si crea un cluster di cache con ElastiCache (Memcached) e si sceglie dual-stack come tipo di rete, è necessario designare un tipo di rilevamento IP, ovvero oppure. IPv4 IPv6 ElastiCache imposterà come impostazione predefinita il tipo di rete e il rilevamento IPIPv6, ma ciò può essere modificato. Se si utilizza Individuazione automatica, solo gli indirizzi IP del tipo IP scelto vengono restituiti al client Memcached. Per ulteriori informazioni, consulta [Identifica automaticamente i nodi del cluster \(Memcached\)](#).

Per mantenere la retrocompatibilità con tutti i client esistenti, viene introdotto il rilevamento IP, che consente di selezionare il tipo di IP (ovvero IPv4 oIPv6) da pubblicizzare nel protocollo di rilevamento. Sebbene ciò limiti il rilevamento automatico a un solo tipo di IP, il dual-stack è comunque vantaggioso per i carichi di lavoro abilitati alla modalità cluster, in quanto consente le migrazioni (o il rollback) da un tipo IP Discovery IPv4 a un tipo di IP IPv6 Discovery senza tempi di inattività.

TLSCluster dual stack abilitati ElastiCache

Quando TLS è abilitato per ElastiCache i cluster, le funzioni di rilevamento dei cluster come `cluster slotscluster shards`, e `cluster nodes` con Valkey o Redis OSS e `config get cluster` con Memcached restituiscono hostnames anziché. IPs I nomi host vengono quindi utilizzati anziché per connettersi al cluster ed IPs eseguire una stretta di mano. ElastiCache TLS Ciò significa che i client non saranno interessati dal parametro Individuazione IP. Per i cluster TLS abilitati, il parametro IP Discovery non ha alcun effetto sul protocollo IP preferito. Invece, il protocollo IP utilizzato sarà determinato dal protocollo IP preferito dal client per la risoluzione dei nomi host. DNS

Per esempi su come configurare una preferenza del protocollo IP durante la risoluzione dei DNS nomi host, vedere. [TLSCluster dual stack ElastiCache abilitati](#)

Utilizzo di AWS Management Console (Valkey e Redis) OSS

Quando crei un cluster utilizzando AWS Management Console, in Connettività, scegli un tipo di rete IPv4, IPv6o Dual stack. Se stai creando un cluster Valkey o Redis OSS (modalità cluster abilitata) e scegli il dual stack, devi selezionare un tipo di Discovery IP, oppure. IPv6 IPv4

Per ulteriori informazioni, consulta [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#) o [Creazione di un Valkey o Redis OSS \(modalità cluster disabilitata\) \(Console\)](#).

Quando si crea un gruppo di replica utilizzando il AWS Management Console, è possibile scegliere un tipo di rete IPv4, IPv6o Dual stack. Se si sceglie il dual stack, è necessario selezionare un tipo di Discovery IP, oppure. IPv6 IPv4

Per ulteriori informazioni, consulta [Creazione da zero di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#) o [Creazione da zero di un gruppo di replica in Valkey o Redis OSS \(modalità cluster abilitata\)](#).

Utilizzo di AWS Management Console (Memcached)

Quando crei un cluster di cache utilizzando AWS Management Console, in Connettività, scegli un tipo di rete IPv4, IPv6o Dual stack. Se si sceglie il dual stack, è necessario selezionare un tipo di Discovery IP, oppureIPv6. IPv4

Per ulteriori informazioni, consulta [Creazione di un cluster Memcached \(console\)](#).

Utilizzo di CLI con Valkey, Redis OSS o Memcached

Redis OSS

Quando si crea un cluster di cache con Valkey o Redis OSS utilizzandoCLI, si utilizza il [create-cache-cluster](#)comando e si specificano i NetworkType parametri and: 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 utilizzando il CLI, si utilizza il [create-replication-group](#) comando e si specificano i NetworkType parametri and: 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 viene utilizzato IPv4 per il rilevamento IP utilizzando il CLI, si utilizza il [create-replication-group](#) comando e si specificano i parametri NetworkType andIPDiscovery:

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 \  
  --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 utilizzato IPv6 per il rilevamento IP utilizzando il CLI, si utilizza il [create-replication-group](#) comando e si specificano i parametri NetworkType andIPDiscovery:

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
```

Memcached

Quando si crea un cluster di cache con Memcached utilizzando il CLI, si utilizza il [create-cache-cluster](#) comando e si specificano i parametri and: NetworkType IPDiscovery

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id "cluster-test" \  
  --engine memcached \  
  --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 memcached ^  
  --cache-node-type cache.m5.large ^  
  --num-cache-nodes 1 ^  
  --network-type dual_stack ^  
  --ip-discovery ipv4
```

Identifica automaticamente i nodi del cluster (Memcached)

Per i cluster che utilizzano il motore Memcached, ElastiCache supporta Auto Discovery, la capacità dei programmi client di identificare automaticamente tutti i nodi di un cluster di cache e di avviare e mantenere le connessioni a tutti questi nodi.

Note

L'Auto Discovery è stato aggiunto per i cluster di cache in esecuzione su Amazon ElastiCache Memcached. Auto Discovery non è disponibile per i motori Valkey o Redis. OSS

Con Individuazione automatica, non occorre collegare manualmente l'applicazione ai singoli nodi di cache; invece, l'applicazione si collega a un nodo Memcached e recupera l'elenco di nodi. Da tale elenco l'applicazione è in grado di conoscere il resto dei nodi nel cluster e può connettersi a uno qualsiasi di essi. Non è necessario codificare gli endpoint dei singoli nodi di cache nell'applicazione.

Se si utilizza un tipo di rete dual stack sul cluster, Auto Discovery restituirà solo IPv6 gli indirizzi IPv4 o, a seconda di quello selezionato. Per ulteriori informazioni, consulta [Scelta del tipo di rete in ElastiCache](#).

Tutti i nodi di cache nel cluster mantengono un elenco di metadati relativi a tutti gli altri nodi. Questi metadati vengono aggiornati ogni volta che si aggiungono o rimuovono nodi dal cluster.

Argomenti

- [Vantaggi dell'individuazione automatica con Memcached](#)
- [Funzionamento di Individuazione automatica](#)
- [Uso di Individuazione automatica](#)
- [Connessione manuale ai nodi di cache Memached](#)
- [Aggiungere Auto Discovery alla libreria client Memcached](#)
- [ElastiCache clienti con rilevamento automatico](#)

Vantaggi dell'individuazione automatica con Memcached

Quando si utilizza Memcached, Auto Discovery offre i seguenti vantaggi:

- Quando si incrementa il numero di nodi in un cluster di cache, i nuovi nodi vengono registrati automaticamente con l'endpoint di configurazione e con tutti gli altri nodi. Quando si rimuovono nodi dal cluster di cache, i nodi che vengono a mancare annullano automaticamente la loro registrazione. In entrambi i casi, tutti gli altri nodi nel cluster vengono aggiornati con i metadati di nodi di cache più recenti.
- Guasti nei nodi di cache vengono rilevati automaticamente; nodi non riusciti vengono sostituiti automaticamente.

Note

Finché la sostituzione del nodo non viene completata, il nodo continuerà a restituire un errore.

- Un programma client deve solo connettersi all'endpoint di configurazione. Successivamente, la libreria di Individuazione automatica si collega a tutti gli altri nodi nel cluster.
- I programmi client eseguono il polling del cluster una volta il minuto (questo intervallo può essere modificato se necessario). In caso di modifiche alla configurazione del cluster, ad esempio nodi nuovi o eliminati, il client riceve un elenco aggiornato di metadati. Quindi il client si collega a, o si scollega da, questi nodi in base alle esigenze.

Auto Discovery è abilitato su tutti i cluster di ElastiCache cache Memcached. Per utilizzare questa caratteristica non è necessario riavviare i nodi di cache.

Funzionamento di Individuazione automatica

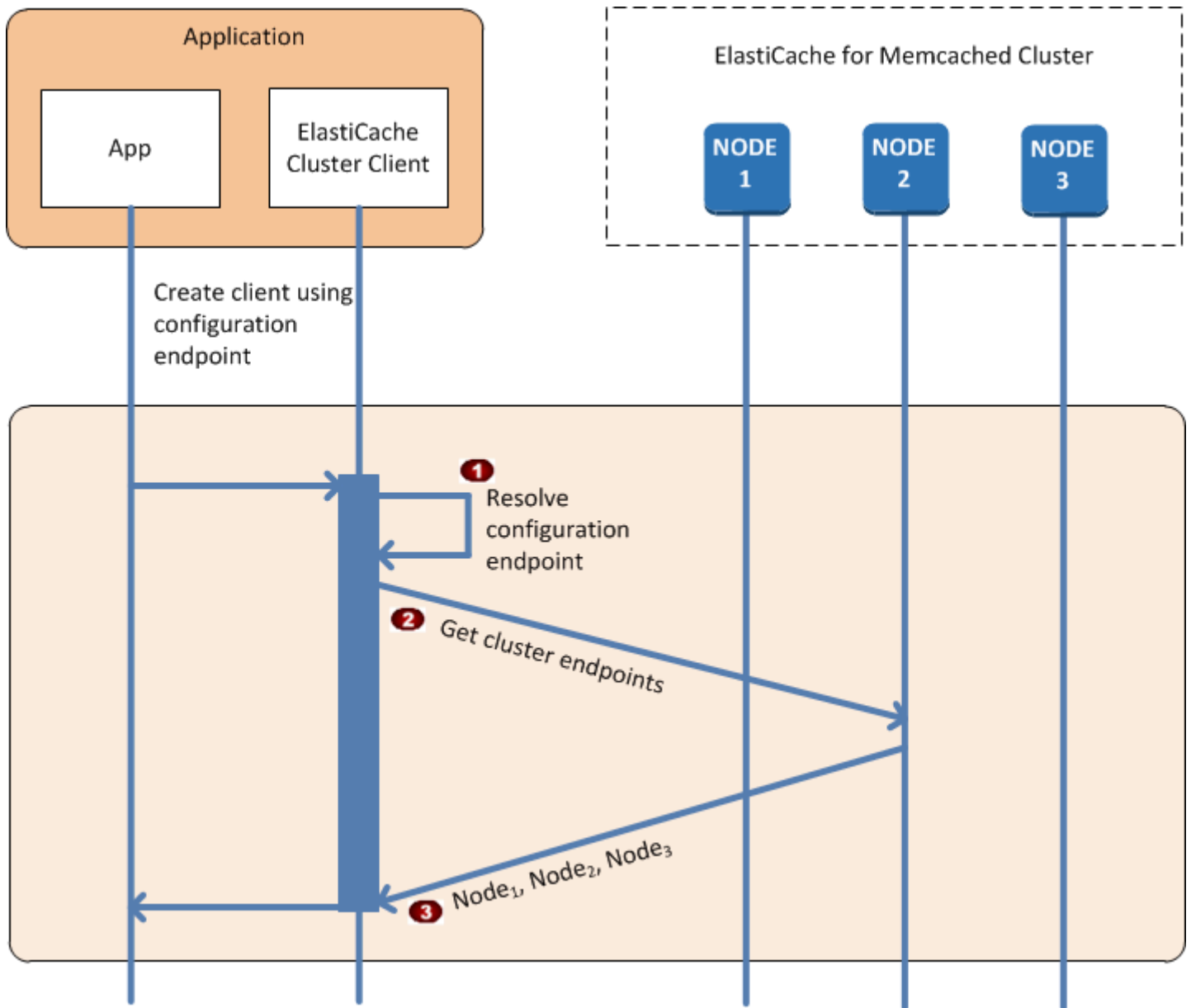
Argomenti

- [Connessione ai nodi di cache](#)
- [Operazioni cluster normali](#)
- [Altre operazioni](#)

Questa sezione descrive come le applicazioni client utilizzano il ElastiCache Cluster Client per gestire le connessioni dei nodi di cache e interagire con gli elementi di dati nella cache.

Connessione ai nodi di cache

Dal punto di vista dell'applicazione, la connessione all'endpoint di configurazione del cluster non è diverso dalla connessione diretta a un singolo nodo di cache. Il diagramma di sequenza seguente mostra il processo di connessione ai nodi di cache.



Processo di connessione ai nodi di cache

- L'applicazione risolve il nome dell'endpoint di configurazione. DNS Poiché l'endpoint di configurazione mantiene le CNAME voci per tutti i nodi della cache, il DNS nome viene risolto in uno dei nodi; il client può quindi connettersi a quel nodo.
- Il client richiede le informazioni di configurazione per tutti gli altri nodi. Poiché ogni nodo mantiene informazioni di configurazione per tutti i nodi nel cluster, qualsiasi nodo può passare informazioni di configurazione al client su richiesta.

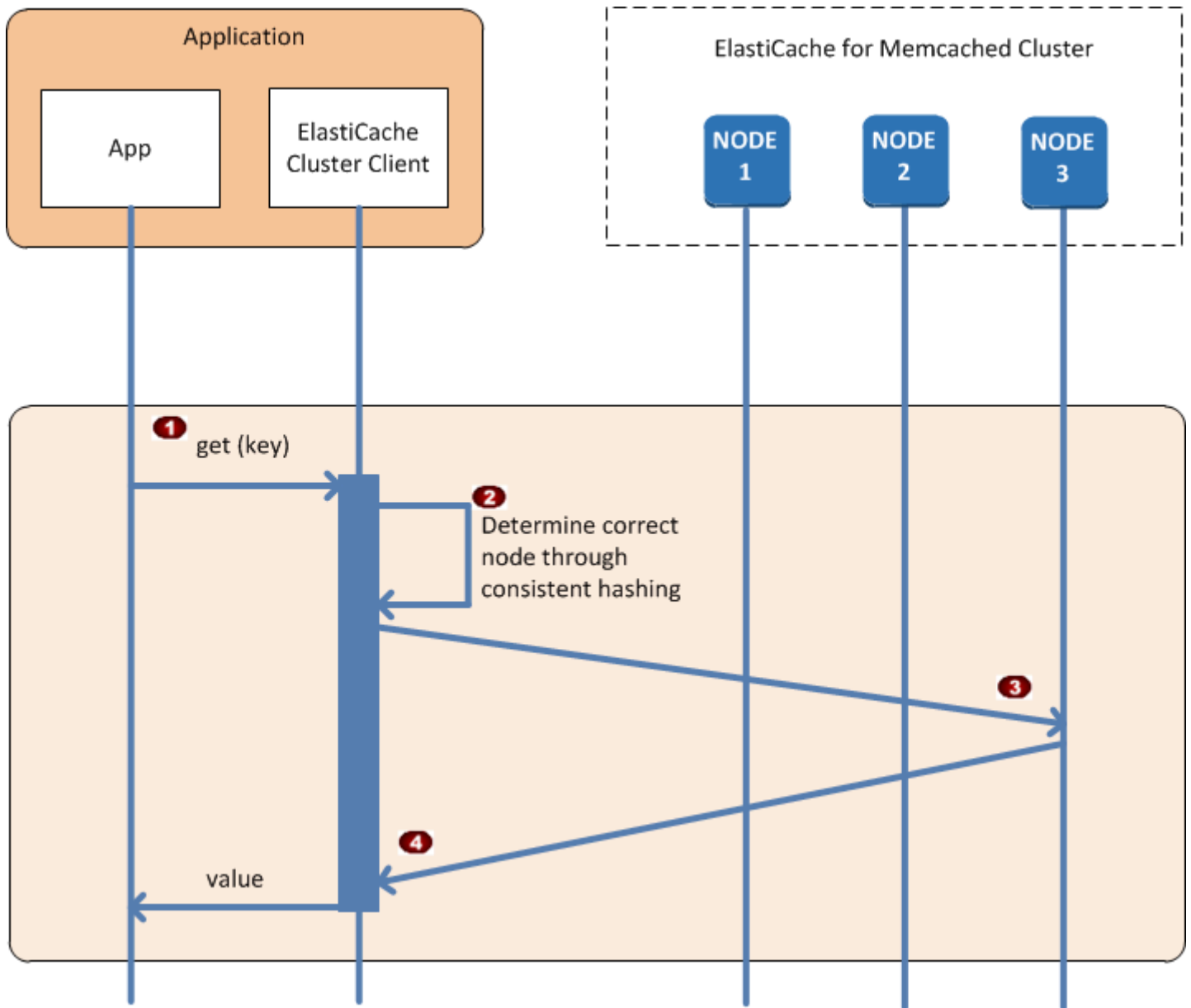
- Il client riceve l'elenco corrente di nomi host e indirizzi IP dei nodi di cache. Può quindi connettersi a tutti gli altri nodi nel cluster.

Note

Il programma client aggiorna il suo elenco di nomi host e indirizzi IP dei nodi di cache una volta il minuto. Questo intervallo di polling può essere modificato se necessario.

Operazioni cluster normali

Quando l'applicazione è connessa a tutti i nodi di cache, ElastiCache Cluster Client determina quali nodi devono memorizzare i singoli elementi di dati e quali nodi devono essere interrogati per tali elementi di dati in un secondo momento. Il diagramma di sequenza seguente mostra il processo di operazioni cluster normali.



Processo di operazioni cluster normali

- L'applicazione emette una richiesta get per un particolare item di dati, identificato dalla sua chiave.
- Il client utilizza un algoritmo di hashing rispetto alla chiave per determinare quale nodo di cache contiene l'item di dati.
- L'item di dati viene richiesto dal nodo appropriato.
- L'item di dati viene restituito all'applicazione.

Altre operazioni

In alcune situazioni, è possibile effettuare una modifica ai nodi di un cluster. Ad esempio, è possibile aggiungere un nodo aggiuntivo per soddisfare una domanda aggiuntiva oppure eliminare un nodo per risparmiare denaro durante i periodi di domanda ridotta. Oppure potresti sostituire un nodo a causa di un errore di un nodo di un tipo o di un altro.

In presenza di una modifica nel cluster che richiede un aggiornamento dei metadati negli endpoint del cluster, tale modifica viene apportata a tutti i nodi contemporaneamente. Pertanto, i metadati in qualsiasi nodo specificato sono coerenti con i metadati in tutti gli altri nodi nel cluster.

In ciascuno di questi casi, i metadati sono coerenti tra tutti i nodi in qualsiasi momento poiché i metadati vengono aggiornati contemporaneamente per tutti i nodi nel cluster. Si consiglia di utilizzare sempre l'endpoint di configurazione per ottenere gli endpoint dei diversi nodi nel cluster. L'uso dell'endpoint di configurazione evita che vengano ricevuti dati dell'endpoint da un nodo che "scompare".

Aggiunta di un nodo

Durante il periodo di attivazione del nodo, il suo endpoint non è incluso nei metadati. Non appena il nodo diventa disponibile, viene aggiunto ai metadati di ciascuno dei nodi del cluster. In questo scenario, i metadati sono coerenti tra tutti i nodi e sarai in grado di interagire con il nuovo nodo solo dopo che questo diventa disponibile. Il nodo non è noto prima che diventi disponibile e l'interazione con i nodi nel cluster avviene come se il nuovo nodo non esistesse.

Eliminazione di un nodo

Quando un nodo viene rimosso, il relativo endpoint viene rimosso innanzitutto dai metadati e quindi il nodo viene rimosso dal cluster. In questo scenario, i metadati in tutti i nodi sono coerenti e in nessun momento conterranno l'endpoint per il nodo che deve essere rimosso mentre il nodo non è disponibile. Durante la rimozione del nodo il tempo non è segnalato nei metadati e pertanto l'applicazione interagisce solo con gli $n-1$ nodi rimanenti, come se il nodo non esistesse.

Sostituzione di un nodo

Se un nodo ElastiCache si guasta, lo rimuove e ne avvia uno sostitutivo. Il processo di sostituzione richiede alcuni minuti. Durante questo periodo di tempo i metadati in tutti i nodi mostrano ancora l'endpoint per il nodo non riuscito, ma qualsiasi tentativo di interagire con il nodo restituisce un errore. Pertanto, la logica deve sempre includere la logica di ripetizione.

Uso di Individuazione automatica

Per iniziare a utilizzare Auto Discovery con ElastiCache (Memcached), procedi nel seguente modo:

- [Ottieni l'endpoint di configurazione](#)
- [Scarica il ElastiCache Cluster Client](#)
- [Modifica il tuo programma applicativo](#)

Ottieni l'endpoint di configurazione

Per connettersi a un cluster, i programmi client devono conoscere l'endpoint di configurazione del cluster. Consulta l'argomento [Ricerca degli endpoint di un cluster \(console\) \(Memcached\)](#)

Puoi anche utilizzare il comando `aws elasticache describe-cache-clusters` con il parametro `--show-cache-node-info`:

Qualunque sia il metodo utilizzato per trovare gli endpoint del cluster, l'indirizzo dell'endpoint di configurazione conterrà sempre `.cfg`.

Example Ricerca degli endpoint utilizzando il comando for AWS CLI ElastiCache

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

Per Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

Questa operazione produce un output simile al seguente (JSONformato):

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {
```

```
    "CacheNodeId": "0001",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.fnjyzo.cfg.0001.use1.cache.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
    "CustomerAvailabilityZone": "us-east-1e"
  },
  {
    "CacheNodeId": "0002",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.fnjyzo.cfg.0002.use1.cache.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
    "CustomerAvailabilityZone": "us-east-1a"
  }
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "Multiple",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-10-12T21:39:28.001Z",
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "available",
"NumCacheNodes": 2,
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
"CacheSubnetGroupName": "default",
"EngineVersion": "1.4.24",
"PendingModifiedValues": {},
"PreferredMaintenanceWindow": "sat:06:00-sat:07:00",
```

```
        "CacheNodeType": "cache.r3.large"  
    }  
]  
}
```

Scarica il ElastiCache Cluster Client

Per sfruttare Auto Discovery, i programmi client devono utilizzare il ElastiCacheCluster Client. Il ElastiCache Cluster Client è disponibile per Java e. PHP NETe contiene tutta la logica necessaria per scoprire e connettersi a tutti i nodi della cache.

Per scaricare il ElastiCache Cluster Client

1. Accedi alla console di AWS gestione e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dalla ElastiCache console, scegli ElastiCache Cluster Client, quindi scegli Scarica.

Il codice sorgente per ElastiCache Cluster Client for Java è disponibile all'indirizzo <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java>. Questa libreria è basata sul client Spymemcached di uso comune. Il ElastiCache Cluster Client è rilasciato con la licenza Amazon Software <https://aws.amazon.com/asl>. Sei libero di modificare il codice fonte come meglio credi. Puoi persino incorporare il codice in altre librerie Memcached open source o nel tuo codice client.

Note

Per utilizzare il ElastiCache Cluster Client perPHP, devi prima installarlo sulla tua EC2 istanza Amazon. Per ulteriori informazioni, consulta [Installazione del client del cluster ElastiCache per PHP](#).

Per un client TLS supportato, scarica il file binario con la PHP versione 7.4 o successiva. Per utilizzare il ElastiCache Cluster Client per. NET, dovrai prima installarlo sulla tua EC2 istanza Amazon. Per ulteriori informazioni, consulta [Installazione del client del ElastiCache cluster per. NET](#).

Modifica il tuo programma applicativo

Modifica il programma dell'applicazione in modo che utilizzi Individuazione automatica. Le seguenti sezioni mostrano come utilizzare il ElastiCache Cluster Client per JavaPHP, e. NET.

⚠ Important

Durante la specifica dell'endpoint di configurazione del cluster, assicurati che l'indirizzo dell'endpoint contenga ".cfg" come mostrato qui. Non utilizzare un endpoint CNAME o senza «.cfg».

```
"mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
```

La mancata specifica esplicita dell'endpoint di configurazione del cluster determina la configurazione di un nodo specifico.

Utilizzo del ElastiCache Cluster Client per Java

Il programma seguente mostra come utilizzare il ElastiCache Cluster Client per connettersi a un endpoint di configurazione del cluster e aggiungere un elemento di dati alla cache. Utilizzando Individuazione automatica, il programma si connette a tutti i nodi nel cluster senza ulteriori interventi.

```
package com.amazon.elasticache;

import java.io.IOException;
import java.net.InetSocketAddress;

// Import the &AWS;-provided library with Auto Discovery support
import net.spy.memcached.MemcachedClient;

public class AutoDiscoveryDemo {

    public static void main(String[] args) throws IOException {

        String configEndpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
        Integer clusterPort = 11211;

        MemcachedClient client = new MemcachedClient(
            new InetSocketAddress(configEndpoint,
                clusterPort));
        // The client will connect to the other cache nodes automatically.

        // Store a data item for an hour.
        // The client will decide which cache host will store this item.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

```
}  
}
```

Utilizzo del ElastiCache Cluster Client per PHP

Il programma seguente mostra come utilizzare il ElastiCache Cluster Client per connettersi a un endpoint di configurazione del cluster e aggiungere un elemento di dati alla cache. Utilizzando Individuazione automatica, il programma si connette a tutti i nodi in un cluster senza ulteriori interventi.

Per utilizzare il ElastiCache Cluster Client perPHP, devi prima installarlo sulla tua EC2 istanza Amazon. Per ulteriori informazioni, consulta [Installazione del client del cluster ElastiCache per PHP](#)

```
<?php  
  
/**  
 * Sample PHP code to show how to integrate with the Amazon ElastiCache  
 * Auto Discovery feature.  
 */  
  
/* Configuration endpoint to use to initialize memcached client.  
 * This is only an example. */  
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";  
  
/* Port for connecting to the ElastiCache cluster.  
 * This is only an example */  
$server_port = 11211;  
  
/**  
 * The following will initialize a Memcached client to utilize the Auto Discovery  
 * feature.  
 *  
 * By configuring the client with the Dynamic client mode with single endpoint, the  
 * client will periodically use the configuration endpoint to retrieve the current  
 * cache  
 * cluster configuration. This allows scaling the cache cluster up or down in number  
 * of nodes  
 * without requiring any changes to the PHP application.  
 *  
 * By default the Memcached instances are destroyed at the end of the request.  
 * To create an instance that persists between requests,  
 * use persistent_id to specify a unique ID for the instance.  
 * All instances created with the same persistent_id will share the same connection. */
```



```
* See http://php.net/manual/en/memcached.construct.php for more information.
*/
$dynamic_client = new Memcached('persistent-id');
$dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
$dynamic_client->addServer($server_endpoint, $server_port);

/**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
$dynamic_client->set('key', 'value', 60);

/**
 * Configuring the client with Static client mode disables the usage of Auto Discovery
 * and the client operates as it did before the introduction of Auto Discovery.
 * The user can then add a list of server endpoints.
 */
$static_client = new Memcached('persistent-id');
$static_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::STATIC_CLIENT_MODE);
$static_client->addServer($server_endpoint, $server_port);

/**
 * Store the data without expiration.
 * The client will decide which cache host will store this item.
 */
$static_client->set('key', 'value');
?>
```

Per un esempio su come utilizzare il ElastiCache Cluster Client con TLS enabled, consulta [Using in transit encryption with PHP and Memcached](#).

Utilizzo del ElastiCache Cluster Client per. NET

Note

La ElastiCache .NET client del cluster è obsoleto a maggio 2022.

.NET client for ElastiCache è open source presso <https://github.com/aws-labs/elasticache-cluster-config-net>.

. NETLe applicazioni in genere ottengono le proprie configurazioni dal proprio file di configurazione. Di seguito viene riportato un file config dell'applicazione di esempio.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section
      name="clusterclient"
      type="Amazon.ElastiCacheCluster.ClusterConfigSettings,
Amazon.ElastiCacheCluster" />
  </configSections>

  <clusterclient>
    <!-- the hostname and port values are from step 1 above -->
    <endpoint hostname="mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
port="11211" />
  </clusterclient>
</configuration>
```

Il programma C# riportato di seguito mostra come utilizzare il ElastiCache Cluster Client per connettersi a un endpoint di configurazione del cluster e aggiungere un elemento di dati alla cache. Utilizzando Individuazione automatica, il programma si connette a tutti i nodi in un cluster senza ulteriori interventi.

```
// *****
// Sample C# code to show how to integrate with the Amazon ElastiCcache Auto Discovery
// feature.

using System;

using Amazon.ElastiCacheCluster;

using Enyim.Caching;
using Enyim.Caching.Memcached;

public class DotNetAutoDiscoveryDemo {

    public static void Main(String[] args) {

        // instantiate a new client.
        ElastiCacheClusterConfig config = new ElastiCacheClusterConfig();
        MemcachedClient memClient = new MemcachedClient(config);
```

```
// Store the data for 3600 seconds (1hour) in the cluster.  
// The client will decide which cache host will store this item.  
memClient.Store(StoreMode.Set, 3600, "This is the data value.");  
  
} // end Main  
  
} // end class DotNetAutoDiscoverDemo
```

Connessione manuale ai nodi di cache Memcached

Se il programma client non utilizza Auto Discovery, può connettersi manualmente a ciascuno dei nodi di cache Memcached. Questo è il comportamento di default per i client Memcached.

Puoi ottenere un elenco di nomi host e numeri porta dei nodi di cache dalla [AWS Console di gestione](#). È inoltre possibile utilizzare il AWS CLI `aws elasticache describe-cache-clusters` comando con il `--show-cache-node-info` parametro.

Example

Il seguente frammento di codice Java mostra come connettersi a tutti i nodi in un cluster di cache di quattro nodi:

```
...  
  
ArrayList<String> cacheNodes = new ArrayList<String>(  
    Arrays.asList(  
        "mycachecluster.fnjyzo.0001.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0002.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0003.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0004.use1.cache.amazonaws.com:11211"));  
  
MemcachedClient cache = new MemcachedClient(AddrUtil.getAddresses(cacheNodes));  
  
...
```

Important

Se aumenti o riduci il cluster di cache aggiungendo o rimuovendo nodi, sarà necessario aggiornare l'elenco di nodi nel codice client.

Aggiungere Auto Discovery alla libreria client Memcached

Le informazioni di configurazione per Auto Discovery sono archiviate in modo ridondante in ogni nodo del cluster di cache Memcached. Le applicazioni client possono eseguire la query di qualsiasi nodo di cache e ottenere le informazioni di configurazione per tutti i nodi nel cluster.

Il modo utilizzato dall'applicazione dipende dalla versione del motore della cache:

- Se la versione del motore della cache è 1.4.14 o successive, utilizza il comando `config`.
- Se la versione del motore della cache è precedente alla 1.4.14, utilizza il comando `get AmazonElastiCache:cluster`.

I risultati di questi due comandi sono identici e vengono descritti nella sezione [Formato di output](#) sottostante.

Versione del motore della cache 1.4.14 o successiva

Per versioni del motore della cache 1.4.14 o successive, utilizza il comando `config`. Questo comando è stato aggiunto ai protocolli Memcached ASCII e binari da ElastiCache, ed è implementato nel Cluster Client. ElastiCache Se desideri utilizzare Individuazione automatica con un'altra libreria client, allora sarà necessario espandere tale libreria per supportare il comando `config`.

Note

La documentazione seguente riguarda il ASCII protocollo; tuttavia, il `config` comando supporta entrambi ASCII i protocolli e il binario. Se desideri aggiungere il supporto Auto Discovery utilizzando il protocollo binario, fai riferimento al [codice sorgente del ElastiCache Cluster Client](#).

Sintassi

```
config [sub-command] [key]
```

Opzioni

Nome	Descrizione	Richiesto
sub-command		Sì

Nome	Descrizione	Richiesto
	Il sottocomando utilizzando per interagire con un nodo di cache. Per Individuazione automatica, questo sottocomando è <code>get</code> .	
<code>key</code>	La chiave sotto cui viene archiviata la configurazione cluster. Per Individuazione automatica, questa chiave è denominata <code>cluster</code> .	Sì

Per ottenere le informazioni di configurazione del cluster, utilizza il seguente comando:

```
config get cluster
```

Versione del motore della cache precedente a 1.4.14

Per ottenere le informazioni di configurazione del cluster, utilizza il seguente comando:

```
get AmazonElastiCache:cluster
```

Note

Non manomettete la chiave «:clusterAmazonElastiCache», poiché è qui che risiedono le informazioni di configurazione del cluster. Se si sovrascrive questa chiave, è possibile che il client venga configurato in modo errato per un breve periodo di tempo (non più di 15 secondi) prima di aggiornare ElastiCache automaticamente e correttamente le informazioni di configurazione.

Formato di output

A prescindere che utilizzi `config get cluster` o `get AmazonElastiCache:cluster`, la risposta è costituita sempre da due righe:

- Il numero di versione delle informazioni di configurazione. Ogni volta che si aggiunge o rimuove un nodo dal cluster di cache, il numero di versione viene incrementato di uno.
- Un elenco di nodi di cache. Ogni nodo nell'elenco è rappresentato da un gruppo `hostname|ip-address|port` e ogni nodo è delimitato da uno spazio.

Alla fine di ciascuna riga vengono visualizzati un carattere di ritorno a capo e un carattere di avanzamento riga (CR + LF). La riga di dati contiene un carattere di avanzamento riga (LF) alla fine, cui viene aggiunto CR + LF. La riga di versione config è terminata da LF senza CR.

Un cluster di cache contenente tre nodi viene rappresentato come riportato di seguito:

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Ogni nodo viene visualizzato sia con l'CNAMEndirizzo IP che con quello privato. CNAMEsarà sempre presente; se l'indirizzo IP privato non è disponibile, non verrà mostrato; tuttavia, i caratteri pipe "|" verranno comunque stampati.

Example

Di seguito è riportato un esempio del payload restituito quando si esegue la query delle informazioni di configurazione:

```
CONFIG cluster 0 136\r\n
12\n
myCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211
myCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\n\r\n
END\r\n
```

Note

- La seconda riga indica che le informazioni di configurazione sono state modificate dodici volte finora.
- Nella terza riga, l'elenco dei nodi è in ordine alfabetico per nome host. Questo ordinamento deve essere in una sequenza diversa da quella attualmente in uso nell'applicazione client.

ElastiCache clienti con rilevamento automatico

I programmi client del cluster possono identificare e connettersi automaticamente a tutti i nodi del cluster di cache che eseguono il motore Memcached.

Questa sezione descrive l'installazione e la configurazione di and. ElastiCache PHP NETclient da utilizzare con auto discovery.

Argomenti

- [Installazione e compilazione di client del cluster](#)
- [Configurazione dei client ElastiCache](#)

Installazione e compilazione di client del cluster

Questa sezione tratta l'installazione, la configurazione e la compilazione di and. PHP NETClient cluster Amazon ElastiCache Auto Discovery.

Argomenti

- [Installazione del client del ElastiCache cluster per. NET](#)
- [Installazione del client del cluster ElastiCache per PHP](#)
- [Compilazione del codice sorgente per il client del ElastiCache cluster per PHP](#)

Installazione del client del ElastiCache cluster per. NET

Puoi trovare il ElastiCache . NETCodice Cluster Client come open source su <https://github.com/awslabs/elasticache-cluster-config-net>.

Questa sezione descrive come installare, aggiornare e rimuovere. NETcomponenti per le EC2 istanze ElastiCache Cluster Client on Amazon. Per ulteriori informazioni sull'individuazione automatica, consulta [Identifica automaticamente i nodi del cluster \(Memcached\)](#). Ad esempio. NETcodice per usare il client, vedi [Utilizzo del ElastiCache Cluster Client per. NET](#).

Argomenti

- [Installazione. NET](#)
- [Scarica il ElastiCache . NETclient cluster per ElastiCache](#)
- [Installa AWS gli assieme con NuGet](#)

Installazione. NET

Devi averlo fatto. NET3.5 o versione successiva installata per utilizzare AWS . NETSDKper ElastiCache. Se non ne hai. NET3.5 o versione successiva, puoi scaricare e installare la versione più recente da <http://www.microsoft.com/net>.

Scarica il ElastiCache . NETclient cluster per ElastiCache

Per scaricare il ElastiCache . NETclient del cluster

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, fai clic su ElastiCache Cluster Client.
3. Nell'elenco Scarica ElastiCache Memcached Cluster Client, selezionare. NET, quindi fate clic su Scarica.

Installa AWS gli assieme con NuGet

NuGet è un sistema di gestione dei pacchetti per. NETpiattaforma. NuGet è consapevole delle dipendenze dell'assieme e installa automaticamente tutti i file richiesti. NuGet gli assembly installati vengono archiviati con la soluzione, anziché in una posizione centrale, ad esempioProgram Files, in modo da poter installare versioni specifiche di un'applicazione senza creare problemi di compatibilità.

Installazione NuGet

NuGet può essere installato dalla Galleria di installazione in poiMSDN; vedere <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>. Se si utilizza Visual NuGet Studio 2010 o versione successiva, viene installato automaticamente.

È possibile utilizzare NuGet da Solution Explorer o dalla console di Package Manager.

Utilizzo NuGet da Solution Explorer

NuGet Da utilizzare da Solution Explorer in Visual Studio 2010

1. Dal menu Strumenti, scegli Gestione pacchetti libreria.
2. Fai clic su Console di Gestione pacchetti.

NuGet Da utilizzare da Solution Explorer in Visual Studio 2012 o Visual Studio 2013

1. Dal menu Strumenti, selezionare NuGet Package Manager.
2. Fai clic su Console di Gestione pacchetti.

Dalla riga di comando, è possibile installare gli assembly utilizzando `Install-Package`, come mostrato di seguito.

```
Install-Package Amazon.ElastiCacheCluster
```

[Per visualizzare una pagina per ogni pacchetto disponibile tramite NuGet, ad esempio gli assembly AWS e.Extensions, visitate il NuGet sito Web all'indirizzo <http://www.nuget.org>. AWS SDK](#) La pagina per ogni pacchetto include una riga di comando di esempio per l'installazione del pacchetto tramite la console e un elenco delle versioni precedenti del pacchetto disponibili tramite NuGet

Per ulteriori informazioni sui comandi di Package Manager Console (Console di Gestione pacchetti), consulta <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>.

Installazione del client del cluster ElastiCache per PHP

In questa sezione viene descritto come installare, aggiornare e rimuovere i componenti PHP per il client del cluster ElastiCache su istanze Amazon EC2. Per ulteriori informazioni sull'individuazione automatica, consulta [Identifica automaticamente i nodi del cluster \(Memcached\)](#). Per il codice PHP di esempio per utilizzare il client, consulta [Utilizzo del ElastiCache Cluster Client per PHP](#).

Argomenti

- [Download del pacchetto di installazione](#)
- [Per utenti con già installata l'estensione php-memcached](#)
- [Fasi di installazione per nuovi utenti](#)
- [Rimozione del client del PHP cluster](#)

Download del pacchetto di installazione

Per assicurarti di utilizzare la versione corretta di ElastiCache Cluster ClientPHP, dovrai sapere quale versione di PHP è installata sulla tua EC2 istanza Amazon. Dovrai anche sapere se la tua EC2 istanza Amazon esegue una versione di Linux a 64 o 32 bit.

Per determinare la PHP versione installata sulla tua EC2 istanza Amazon

- Dal prompt dei comandi, eseguire il seguente comando:

```
php -v
```

La PHP versione verrà mostrata nell'output, come in questo esempio:

```
PHP 5.4.10 (cli) (built: Jan 11 2013 14:48:57)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
```

Note

Se la tua versione PHP e quella di Memcached sono incompatibili, riceverai un messaggio di errore simile al seguente:

```
PHP Warning: PHP Startup: memcached: Unable to initialize module
Module compiled with module API=20100525
```

```
PHP compiled with module API=20131226
These options need to match
in Unknown on line 0
```

In tal caso, occorre compilare il modulo dal codice fonte. Per ulteriori informazioni, consulta [Compilazione del codice sorgente per il client del ElastiCache cluster per PHP](#).

Per determinare la tua EC2 AMI architettura Amazon (64 bit o 32 bit)

1. Accedi a AWS Management Console e apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nell'elenco Istanze, fai clic sulla tua EC2 istanza Amazon.
3. Nella scheda Descrizione, cerca il campo AMI:. Un'istanza a 64 bit deve includere x86_64 come parte della descrizione; per un'istanza a 32 bit, cercare i386 o i686 in questo campo.

Ora sei pronto per scaricare il ElastiCache Cluster Client.

Per scaricare il client del ElastiCache cluster per PHP

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dalla ElastiCache console, scegli ElastiCache Cluster Client.
3. Dall'elenco Scarica ElastiCache Memcached Cluster Client, scegli il ElastiCache Cluster Client che corrisponde alla tua PHP versione e AMI architettura, quindi scegli il pulsante Download.

Per utenti con già installata l'estensione php-memcached

Per aggiornare l'installazione **php-memcached**

1. Rimuovere l'installazione precedente dell'estensione Memcached per PHP come descritto dall'argomento [Rimozione del client del PHP cluster](#).
2. Installare la nuova estensione ElastiCache di php-memcached come descritto in precedenza in [Fasi di installazione per nuovi utenti](#).

Fasi di installazione per nuovi utenti

Argomenti

- [Installazione della PHP versione 7.x per i nuovi utenti](#)
- [Installazione di PHP 5.x per nuovi utenti](#)

Installazione della PHP versione 7.x per i nuovi utenti

Argomenti

- [Per installare PHP 7 su un server Ubuntu 14.04 LTS AMI \(64 bit e 32 bit\)](#)
- [Per installare PHP 7 su Amazon Linux 201609 AMI](#)
- [Per installare PHP 7 su SUSE Linux AMI](#)

Per installare PHP 7 su un server Ubuntu 14.04 LTS AMI (64 bit e 32 bit)

1. Avvia una nuova istanza da AMI
2. Esegui i comandi seguenti:

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. Installa PHP 7.

```
sudo yum install php70
```

4. Scarica Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. Estrarre latest-64bit.

```
tar -zxvf latest-64bit
```

6. Con autorizzazioni root, copiare il file dell'artefatto estratto `amazon-elasticache-cluster-client.so` in `/usr/lib/php/20151012`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. Inserire la riga `extension=amazon-elasticache-cluster-client.so` nel file `/etc/php/7.0/cli/php.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php/7.0/cli/php.ini
```

8. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Per installare PHP 7 su Amazon Linux 201609 AMI

1. Avvia una nuova istanza da AMI
2. Esegui il comando seguente:

```
sudo yum install gcc-c++
```

3. Installa PHP 7.

```
sudo yum install php70
```

4. Scarica Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. Estrarre `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Con autorizzazione `root`, copiare il file dell'artefatto estratto `amazon-elasticache-cluster-client.so` in `/usr/lib64/php/7.0/modules/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. Crea il file `50-memcached.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php-7.0.d/50-memcached.ini
```

8. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Per installare PHP 7 su SUSE Linux AMI

1. Avvia una nuova istanza daAMI.
2. Esegui il comando seguente:

```
sudo zypper install gcc
```

3. Installa PHP 7.

```
sudo yum install php70
```

4. Scarica Amazon ElastiCache Cluster Client.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/  
latest-64bit
```

5. Estrarre latest-64bit.

```
tar -zxvf latest-64bit
```

6. Con autorizzazione root, copiare il file dell'artefatto estratto `amazon-elasticache-cluster-client.so` in `/usr/lib64/php7/extensions/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. Inserire la riga `extension=amazon-elasticache-cluster-client.so` nel file `/etc/php7/cli/php.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php7/cli/php.ini
```

8. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Installazione di PHP 5.x per nuovi utenti

Argomenti

- [Per installare PHP 5 su Amazon Linux AMI 2014.03 \(64 bit e 32 bit\)](#)
- [Per installare PHP 5 su Red Hat Enterprise Linux 7.0 \(64-bit e 32-bit\) AMI](#)
- [Per installare PHP 5 su un server Ubuntu 14.04 LTS AMI \(64 bit e 32 bit\)](#)
- [Per installare PHP 5 for SUSE Linux enterprise server 11 \(64 bit o 32 bit\) AMI](#)
- [Altre distribuzioni Linux](#)

Per installare PHP 5 su Amazon Linux AMI 2014.03 (64 bit e 32 bit)

1. Avviare un'istanza Amazon Linux (a 64 bit o 32 bit) ed eseguire l'accesso.
2. Installa le dipendenze: PHP

```
sudo yum install gcc-c++ php php-pear
```

3. Scarica il php-memcached pacchetto corretto per la tua EC2 istanza e PHP versione di Amazon. Per ulteriori informazioni, consulta [Download del pacchetto di installazione](#).
4. Installare php-memcached. URI dovrebbe essere il percorso di download del pacchetto di installazione:

```
sudo pecl install <package download path>
```

Ecco un esempio di comando di installazione per Linux PHP 5.4 a 64 bit. In questo esempio, sostituisci *X.Y.Z* con il numero di versione effettivo:

```
sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```


Note

Assicurati di utilizzare la versione più recente dell'artefatto di installazione.

5. Con il permesso root/sudo, aggiungi un nuovo file denominato `memcached.ini` nella `/etc/php.d` directory e inserisci «`extension=amazon-elasticache-cluster-client.so`» nel file:

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php.d/memcached.ini
```

6. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Per installare PHP 5 su Red Hat Enterprise Linux 7.0 (64-bit e 32-bit) AMI

1. Avviare un'istanza Red Hat Enterprise Linux (64 bit o 32 bit) ed eseguire l'accesso.
2. Installa PHP le dipendenze:

```
sudo yum install gcc-c++ php php-pear
```

3. Scarica il `php-memcached` pacchetto corretto per la tua EC2 istanza e PHP versione di Amazon. Per ulteriori informazioni, consulta [Download del pacchetto di installazione](#).
4. Installare `php-memcached`. URI Dovrebbe essere il percorso di download del pacchetto di installazione:

```
sudo pecl install <package download path>
```

5. Con autorizzazione root/sudo, aggiungere un nuovo file denominato `memcached.ini` nella directory `/etc/php.d` e inserire `extension=amazon-elasticache-cluster-client.so` nel file.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php.d/memcached.ini
```

6. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Per installare PHP 5 su un server Ubuntu 14.04 LTS AMI (64 bit e 32 bit)

1. Avviare un'istanza Ubuntu Linux (a 64 bit o 32 bit) ed eseguire l'accesso.
2. Installa PHP le dipendenze:

```
sudo apt-get update  
sudo apt-get install gcc g++ php5 php-pear
```

3. Scarica il php-memcached pacchetto corretto per la tua EC2 istanza e PHP versione di Amazon. Per ulteriori informazioni, consulta [Download del pacchetto di installazione](#).
4. Installare php-memcached. URI Dovrebbe essere il percorso di download del pacchetto di installazione.

```
sudo pecl install <package download path>
```

Note

Questa fase di installazione consente di installare l'artefatto di compilazione `amazon-elasticache-cluster-client.so` nella directory `/usr/lib/php5/20121212*`. Verifica il percorso assoluto dell'artefatto di compilazione perché è richiesto nella fase successiva.

Se il comando precedente non funziona, è necessario estrarre manualmente l'elemento del PHP client `amazon-elasticache-cluster-client.so` dal `*.tgz` file scaricato e copiarlo nella `/usr/lib/php5/20121212*` directory.

```
tar -xvf <package download path>  
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Con l'autorizzazione `root/sudo`, aggiungi un nuovo file denominato `memcached.ini` nella `/etc/php5/cli/conf.d` directory e inserisci «`extension=<absolute path to .so>`» nel file. `amazon-elasticache-cluster-client`

```
echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee --append /etc/php5/cli/conf.d/memcached.ini
```

6. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Per installare PHP 5 for SUSE Linux enterprise server 11 (64 bit o 32 bit) AMI

1. Avvia un'istanza SUSE Linux (a 64 o 32 bit) e accedi ad essa.
2. Installa le PHP dipendenze:

```
sudo zypper install gcc php53-devel
```

3. Scarica il php-memcached pacchetto corretto per la tua EC2 istanza e PHP versione di Amazon. Per ulteriori informazioni, consulta [Download del pacchetto di installazione](#).
4. Installare php-memcached. URI Dovrebbe essere il percorso di download del pacchetto di installazione.

```
sudo pecl install <package download path>
```

5. Con autorizzazione root/sudo, aggiungere un nuovo file denominato memcached.ini nella directory /etc/php5/conf.d e inserire **extension=amazon-elasticache-cluster-client.so** nel file.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php5/conf.d/memcached.ini
```

6. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Note

Se la Fase 5 non funziona per una delle piattaforme precedenti, verifica il percorso di installazione di `amazon-elasticache-cluster-client.so`. Specifica inoltre il percorso completo del file binario nell'estensione. Inoltre, verifica che quella PHP in uso sia una versione supportata. Sono supportate le versioni da 5.3 a 5.5.

Altre distribuzioni Linux

Su alcuni sistemi, in particolare Cent OS7 e Red Hat Enterprise Linux (RHEL) 7.1, `libsasl2.so.3` è stata sostituita. `libsasl2.so.2` Su questi sistemi, quando si carica il client del ElastiCache cluster, questo tenta e fallisce di trovare e caricare. `libsasl2.so.2` Per risolvere questo problema, crea un collegamento simbolico a `libsasl2.so.3` in modo che quando il client tenta di caricare `libsasl2.so.2`, viene reindirizzato a `libsasl2.so.3`. Il codice seguente crea questo collegamento simbolico:

```
cd /usr/lib64
sudo ln libsasl2.so.3 libsasl2.so.2
```

Rimozione del client del PHP cluster

Argomenti

- [Rimozione di una versione precedente di PHP 7](#)
- [Rimozione di una versione precedente di 5 PHP](#)

Rimozione di una versione precedente di PHP 7

Per rimuovere una versione precedente di PHP 7

1. Rimuovete il `amazon-elasticache-cluster-client.so` file dalla directory PHP lib appropriata come indicato in precedenza nelle istruzioni di installazione. Per informazioni sull'installazione consulta la sezione in [Per utenti con già installata l'estensione php-memcached](#).
2. Rimuovi la riga `extension=amazon-elasticache-cluster-client.so` dal file `php.ini`.
3. Avviare o riavviare il server Apache.

```
sudo /etc/init.d/httpd start
```

Rimozione di una versione precedente di 5 PHP

Per rimuovere una versione precedente di PHP 5

1. Rimuovi l'estensione php-memcached:

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Rimuovi il file `memcached.ini` aggiunto nella directory appropriata come indicato nelle fasi di installazione precedenti.

Compilazione del codice sorgente per il client del ElastiCache cluster per PHP

Questa sezione spiega come ottenere e compilare il codice sorgente per il ElastiCache Cluster Client for. PHP

[Ci sono due pacchetti da cui estrarre GitHub e compilare; aws-elasticache-cluster-client-libmemcached e -. aws-elasticache-cluster-client memcached-for-php](#)

Argomenti

- [Compilazione della libreria libmemcached](#)
- [Compilazione del client di auto ElastiCache discovery Memcached per PHP](#)

Compilazione della libreria libmemcached

Per compilare la libreria -libmemcached aws-elasticache-cluster-client

1. Avvia un'EC2istanza Amazon.
2. Installare le dipendenze di libreria.
 - Su Amazon Linux 2015/09 AMI

```
sudo yum install gcc gcc-c++ autoconf libevent-devel
```

- Su Ubuntu 14.04 AMI

```
sudo apt-get update
sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev
```

3. Estrarre il repository e compilare il codice.

Download and install <https://github.com/aws-labs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz>

Compilazione del client di auto ElastiCache discovery Memcached per PHP

Le seguenti sezioni descrivono come compilare il Memcached Auto Discovery Client ElastiCache

Argomenti

- [Compilazione del client Memcached per ElastiCache 7 PHP](#)

- [Compilazione del client Memcached per 5 ElastiCache PHP](#)

Compilazione del client Memcached per ElastiCache 7 PHP

Esegui il set di comandi seguente nella directory del codice.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
git checkout php7
sudo yum install php70-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-memcached-sasl
make
make install
```

Note

È possibile collegare staticamente la libreria libmemcached al PHP binario in modo che possa essere trasferita su varie piattaforme Linux. A questo scopo, esegui il comando seguente prima di make:

```
sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
```

Compilazione del client Memcached per 5 ElastiCache PHP

Compila `aws-elasticache-cluster-client-memcached-for-php` eseguendo i comandi seguenti nella cartella `aws-elasticache-cluster-client-memcached-for-php/`.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory>
make
make install
```


Configurazione dei client ElastiCache

Un ElastiCache cluster è conforme al protocollo con Valkey, Redis e Memcached. OSS Il codice, le applicazioni e gli strumenti più diffusi che utilizzate oggi con l'ambiente esistente funzioneranno perfettamente con il servizio.

Questa sezione illustra considerazioni specifiche per la connessione ai nodi di cache in ElastiCache

Argomenti

- [Comandi limitati](#)
- [Individuazione di endpoint e numeri di porta dei nodi](#)
- [Connessione per l'utilizzo di Individuazione automatica](#)
- [Connessione ai nodi in un cluster Valkey o Redis OSS](#)
- [DNSnomi e IP sottostante](#)

Comandi limitati

Per offrire un'esperienza di servizio gestito, ElastiCache limita l'accesso a determinati comandi specifici del motore di cache che richiedono privilegi avanzati. Per cluster di cache che eseguono Redis, i seguenti comandi non sono disponibili:

- bgrewriteaof
- bgsave
- config
- debug
- migrate
- replicaof
- save
- slaveof
- shutdown
- sync

Individuazione di endpoint e numeri di porta dei nodi

Per effettuare la connessione a un nodo di cache, l'applicazione deve conoscere l'endpoint e il numero di porta per tale nodo.

Individuazione di endpoint e numeri di porta dei nodi (console)

Per determinare gli endpoint e i numeri di porta dei nodi

1. Accedi alla [console di ElastiCache gestione Amazon](#) e scegli il motore in esecuzione sul tuo cluster.

Viene visualizzato un elenco di tutti i cluster che eseguono il motore scelto.

2. Proseguire di seguito per il motore e la configurazione in esecuzione.
3. Scegliere il nome del cluster di interesse.
4. Individuare le colonne Port (Porta) ed Endpoint per il nodo desiderato.

Individuazione di endpoint e numeri di porta dei nodi di cache (AWS CLI)

Per determinare gli endpoint e i numeri di porta dei nodi di cache, utilizza il comando `describe-cache-clusters` con il parametro `--show-cache-node-info`.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

I DNS nomi completi e i numeri di porta si trovano nella sezione Endpoint dell'output.

Ricerca degli endpoint e dei numeri di porta dei nodi di cache () ElastiCache API

Per determinare gli endpoint e i numeri di porta dei nodi di cache, utilizza l'operazione `DescribeCacheClusters` con il parametro `ShowCacheNodeInfo=true`.

Example

```
https://elasticache.us-west-2.amazonaws.com /  
?Action=DescribeCacheClusters  
&ShowCacheNodeInfo=true  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20140421T220302Z
&Version=2014-09-30
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20140421T220302Z
&X-Amz-Expires=20140421T220302Z
&X-Amz-Signature=<signature>
&X-Amz-SignedHeaders=Host
```

Connessione per l'utilizzo di Individuazione automatica

Se le applicazioni utilizzano Individuazione automatica, è sufficiente conoscere l'endpoint di configurazione per il cluster, anziché i singoli endpoint per ogni nodo di cache. Per ulteriori informazioni, consulta [Identifica automaticamente i nodi del cluster \(Memcached\)](#).

Note

Al momento, Individuazione automatica è disponibile solo per cluster di cache che eseguono Memcached.

Connessione ai nodi in un cluster Valkey o Redis OSS

Note

Al momento, i cluster (API/CLI: gruppi di replica) che supportano la replica e le repliche di lettura sono supportati solo per i cluster che eseguono Valkey o Redis. OSS

Per i cluster, ElastiCache fornisce console e API interfacce per ottenere informazioni di connessione per CLI i singoli nodi.

Per attività di sola lettura, le applicazioni possono connettersi a qualsiasi nodo nel cluster. Tuttavia, per le attività di scrittura, consigliamo che le applicazioni si connettano all'endpoint principale (Valkey o Redis OSS (modalità cluster disabilitata)) o all'endpoint di configurazione (Valkey o Redis OSS (modalità cluster abilitata)) del cluster anziché connettersi direttamente a un nodo. In questo modo, l'applicazione può sempre trovare il nodo corretto, anche se si sceglie di riconfigurare il cluster promuovendo una replica di lettura al ruolo primario.

Connessione a cluster in un gruppo di replica (Console)

Per determinare gli endpoint e i numeri di porta

- Consulta l'argomento [Individuazione degli endpoint di un cluster Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)

Connessione a cluster in un gruppo di replica (AWS CLI)

Per determinare gli endpoint e i numeri di porta dei nodi di cache

Utilizzare il comando `describe-replication-groups` con il nome del gruppo di replica:

```
aws elasticache describe-replication-groups redis2x2
```

Questo comando dovrebbe generare un output simile al seguente:

```
{
  "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": "redis2x2-0001-001"
            },
            {
              "PreferredAvailabilityZone": "us-west-2a",
              "CacheNodeId": "0001",
              "CacheClusterId": "redis2x2-0001-002"
            }
          ]
        }
      ]
    },
    {
      "Status": "available",
```

```

        "Slots": "8192-16383",
        "NodeGroupId": "0002",
        "NodeGroupMembers": [
            {
                "PreferredAvailabilityZone": "us-west-2b",
                "CacheNodeId": "0001",
                "CacheClusterId": "redis2x2-0002-001"
            },
            {
                "PreferredAvailabilityZone": "us-west-2a",
                "CacheNodeId": "0001",
                "CacheClusterId": "redis2x2-0002-002"
            }
        ]
    },
    "ConfigurationEndpoint": {
        "Port": 6379,
        "Address": "redis2x2.9dcv5r.clustercfg.usw2.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "redis2x2",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "13:00-14:00",
    "MemberClusters": [
        "redis2x2-0001-001",
        "redis2x2-0001-002",
        "redis2x2-0002-001",
        "redis2x2-0002-002"
    ],
    "CacheNodeType": "cache.m3.medium",
    "PendingModifiedValues": {}
}
]
}

```

Connessione ai cluster in un gruppo di replica () ElastiCache API

Per determinare gli endpoint e i numeri di porta dei nodi di cache

Chiama `DescribeReplicationGroups` con il seguente parametro:

`ReplicationGroupId` = il nome del gruppo di replica.

Example

```
https://elasticache.us-west-2.amazonaws.com /
?Action=DescribeCacheClusters
&ReplicationGroupId=repgroup01
&Version=2014-09-30
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20140421T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20140421T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20140421T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

DNS nomi e IP sottostante

I client mantengono un elenco di server contenente gli indirizzi e i numeri di porta dei server che detengono i dati di cache. Quando viene utilizzato ElastiCache, DescribeCacheClusters API (o l'utilità della riga di describe-cache-clusters comando) restituisce una DNS voce e un numero di porta completi che possono essere utilizzati per l'elenco dei server.

Important

È importante che le applicazioni client siano configurate per risolvere frequentemente DNS i nomi dei nodi di cache quando tentano di connettersi a un endpoint del nodo di cache.

VPC Installazioni

ElastiCache assicura che sia il DNS nome che l'indirizzo IP del nodo di cache rimangano gli stessi quando i nodi di cache vengono ripristinati in caso di errore.

Non VPC installazioni

ElastiCache assicura che il DNS nome di un nodo di cache rimanga invariato quando i nodi di cache vengono ripristinati in caso di errore; tuttavia, l'indirizzo IP sottostante del nodo di cache può cambiare.

La maggior parte delle librerie client supportano connessioni ai nodi di cache persistenti per impostazione predefinita. Si consiglia di utilizzare connessioni persistenti ai nodi di cache durante

l'utilizzo ElastiCache. La memorizzazione nella DNS cache lato client può avvenire in più posizioni, tra cui le librerie client, il runtime del linguaggio o il sistema operativo client. È opportuno rivedere la configurazione dell'applicazione a ogni livello per essere certi di risolvere frequentemente indirizzi IP per i nodi di cache.

Suddivisione dei dati su più livelli in ElastiCache

ElastiCache con OSS i cluster Valkey o Redis che comprendono un gruppo di replica e utilizzano un tipo di nodo della famiglia r6gd, i dati vengono distribuiti su più livelli tra memoria e storage locale (unità a stato solido). SSD Il data tiering offre una nuova opzione in termini di rapporto prezzo/prestazioni per i OSS carichi di lavoro Valkey o Redis utilizzando unità a stato solido a basso costo () in ogni nodo del cluster oltre all'archiviazione dei dati in memoria. SSDs È ideale per i carichi di lavoro che accedono regolarmente fino al 20 percento del set di dati complessivo e per le applicazioni che possono tollerare una latenza aggiuntiva durante l'accesso ai dati su. SSD

Nei ElastiCache cluster con suddivisione dei dati su più livelli, ElastiCache monitora l'ultimo orario di accesso di ogni elemento archiviato. Quando la memoria disponibile (DRAM) è completamente consumata, ElastiCache utilizza un algoritmo utilizzato meno di recente (LRU) per spostare automaticamente gli elementi a cui si accede meno frequentemente dalla memoria a. SSD Quando successivamente SSD si accede ai dati su, 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.

Si noti che quando si utilizza il tiering dei dati, le chiavi stesse rimangono sempre in memoria, mentre LRU regolano il posizionamento dei valori nella memoria rispetto al 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 da 500 byte, puoi aspettarti in media 300 microsecondi di latenza aggiuntivi per le richieste ai dati archiviati rispetto alle richieste ai dati in memoria. SSD

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 Valkey o Redis supportati in. OSS 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 data tiering è ideale per i carichi di lavoro che accedono regolarmente fino al 20% del set di dati complessivo e per le applicazioni che possono tollerare una latenza aggiuntiva durante l'accesso ai dati su SSD.
- Quando si utilizza la SSD capacità disponibile su nodi a più livelli di dati, è consigliabile che la dimensione del valore sia maggiore della dimensione della chiave. Quando gli elementi vengono spostati tra DRAM e SSD, le chiavi rimarranno sempre in memoria e solo i valori vengono spostati nel 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.
- È necessario utilizzare un motore Valkey 7.2 o successivo oppure Redis OSS 6.2 o successivo.
- 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 [Ridimensionamento ElastiCache](#).

- La scalabilità automatica è supportata sui cluster che utilizzano la suddivisione in più livelli dei dati per Valkey versione 7.2 e successive e Redis versione 7.0.7 e successive. OSS Per ulteriori informazioni, consulta [Auto Scaling dei cluster Valkey e Redis OSS](#)
- Il tiering di dati supporta solo policy maxmemory volatile-lru, allkeys-lru, volatile-lfu, allkeys-lfu e noeviction.
- Il salvataggio senza forkless è supportato per Valkey versione 7.2 e successive e Redis versione 7.0.7 e successive. OSS Per ulteriori informazioni, consulta [Modalità di implementazione di sincronizzazione e backup](#).
- Gli elementi di dimensioni superiori a 128 MiB non vengono spostati in. SSD

Prezzi

I nodi R6gd hanno una capacità totale (memoria+SSD) 4,8 volte superiore e possono aiutarti a ottenere risparmi di oltre il 60% quando funzionano al massimo utilizzo rispetto ai nodi R6g (solo memoria). Per ulteriori [ElastiCache informazioni](#), consulta la pagina dei prezzi.

Monitoraggio

ElastiCache offre metriche progettate specificamente per monitorare i cluster di prestazioni che utilizzano il tiering dei dati. Per monitorare il rapporto tra gli articoli DRAM rispetto aSSD, puoi utilizzare la CURRITEMS metrica di [Metrics](#) for Valkey e Redis. OSS Puoi calcolare la percentuale come: $(\text{CurrItems con Dimension: Tier} = \text{Memory} * 100) / (\text{CurrItems senza filtro dimensionale})$.

Se la politica di sfratto configurata lo consente, ElastiCache inizierà a sfrattare gli articoli quando la percentuale di elementi in memoria scende al di sotto del 5%. Sui nodi configurati con nessuna politica di espulsione, le operazioni di scrittura riceveranno un errore di esaurimento della memoria.

Si consiglia comunque di prendere in considerazione la scalabilità orizzontale per i cluster abilitati alla modalità cluster o la scalabilità verticale per i cluster disabilitati in modalità cluster quando la percentuale di elementi in memoria scende al di sotto del 5%. Per ulteriori informazioni sulla scalabilità, vedere. [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#) Per ulteriori informazioni sulle metriche per i OSS cluster Valkey o Redis che utilizzano il tiering dei dati, consulta. [Metriche per Valkey e Redis OSS](#)

Utilizzo del tiering di dati

Utilizzo del tiering dei dati 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 per Valkey o Redis OSS](#).

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 OSS 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 OSS 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 OSS 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 ().AWS CLI ElastiCache API 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 a AWS Management Console e apri la ElastiCache console 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 Restore (Ripristina).

5. Completare la finestra di dialogo Restore Cluster (Ripristina cluster). Assicurarsi di completare tutti i campi Obbligatori e gli eventuali altri che si desidera modificare rispetto alle impostazioni predefinite.
 1. Cluster ID (ID cluster) : Obbligatorio. Il nome del nuovo cluster.
 2. Modalità cluster abilitata (scalabilità orizzontale): scegli questa opzione per un cluster Valkey o Redis OSS (modalità cluster abilitata).
 3. Tipo di nodo – Specificare `cache.r6gd.xlarge` o qualsiasi altro tipo di nodo della famiglia `r6gd`.
 4. Numero di shard: scegli il numero di shard che desideri 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. Scegli un VPC: scegli il tipo VPC in cui creare questo cluster.
 10. Gruppo di parametri: scegli un gruppo di parametri che riserbi memoria sufficiente per l'OSSoverhead di Valkey o 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 per Valkey o Redis OSS](#).

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 OSS 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 OSS 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 OSS 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 in ElastiCache

Di seguito, puoi trovare le istruzioni sulla creazione di un cluster utilizzando la ElastiCache console, il AWS CLI, o il ElastiCache API.

È inoltre possibile creare un ElastiCache cluster utilizzando [AWS CloudFormation](#). Per ulteriori informazioni, consulta [AWS::ElastiCache::CacheCluster](#) nella Guida per l'utente di AWS Cloud Formation, che include indicazioni su come implementare questo 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

- [Determinazione dei requisiti ElastiCache del cluster](#)
- [Scelta delle dimensioni dei nodi](#)

Determinazione dei requisiti ElastiCache del cluster

Preparazione

Conoscere le risposte alle seguenti domande aiuta a semplificare la creazione ElastiCache 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](#).

- Lancerai il tuo cluster in un cloud privato virtuale (VPC) basato su AmazonVPC?

Important

Se intendi avviare il cluster in unVPC, assicurati di creare un gruppo di sottoreti nello stesso VPC prima di iniziare a creare un cluster. Per ulteriori informazioni, consulta [Sottoreti e gruppi di sottoreti](#).

ElastiCache è progettato per essere accessibile dall'interno AWS tramite AmazonEC2. Tuttavia, se esegui l'avvio in un VPC ambiente basato su Amazon VPC e il cluster si trova in un VPC ambiente, puoi fornire l'accesso dall'esterno AWS. Per ulteriori informazioni, consulta [Accesso alle ElastiCache risorse 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 ElastiCache parametri](#).

Se utilizzi Valkey o RedisOSS, valuta la possibilità di impostare `reserved-memory` o `reserved-memory-percent`. Per ulteriori informazioni, consulta [Gestione della memoria riservata per Valkey e Redis OSS](#).

- Devi creare il tuo gruppo di VPCsicurezza?

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

- Come intendi implementare la tolleranza ai guasti?

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

Argomenti

- [ElastiCache requisiti di memoria e processore](#)
- [Configurazione del cluster Memcached](#)
- [Configurazione dei cluster Valkey e Redis OSS](#)
- [ElastiCache requisiti di scalabilità](#)
- [ElastiCache requisiti di accesso](#)
- [Requisiti per regione, zona di disponibilità e zona locale per ElastiCache](#)

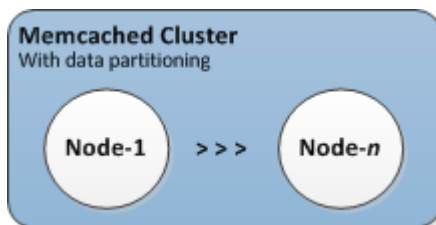
ElastiCache requisiti di memoria e processore

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.

Memcached è un motore multi-thread, quindi il numero di core di un nodo influisce sulla potenza di elaborazione disponibile per il cluster.

Configurazione del cluster Memcached

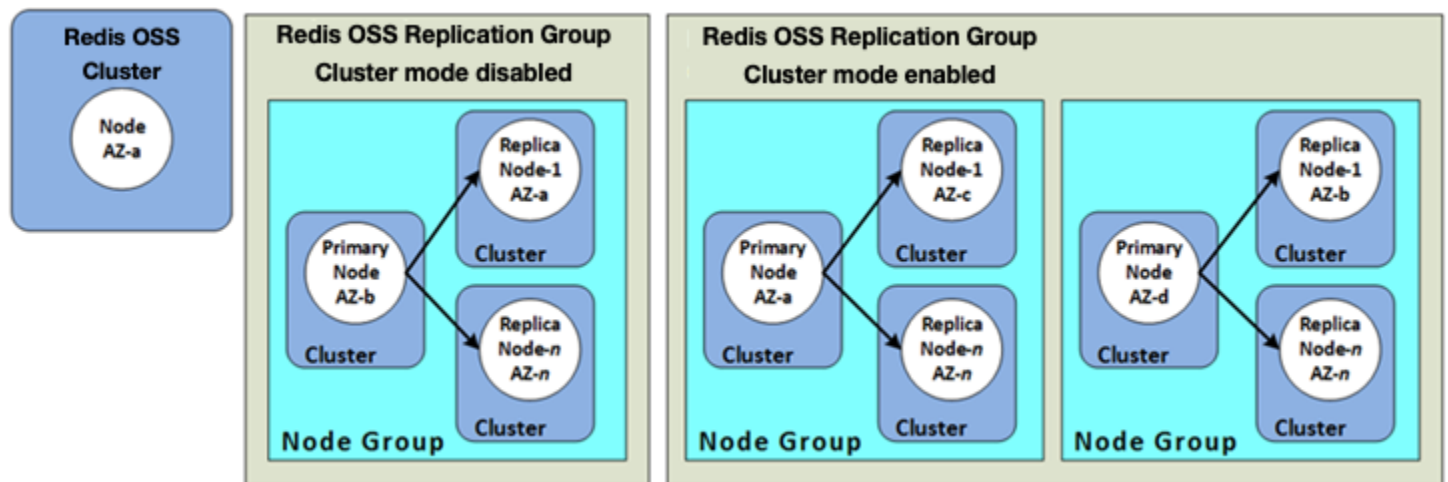
ElastiCache I cluster (Memcached) sono composti da 1 a 60 nodi. I dati presenti in un cluster Memcached vengono partizionati nei nodi del cluster. La tua applicazione si connette a un cluster Memcached tramite un indirizzo di rete chiamato endpoint. Ogni nodo di un cluster Memcached ha un proprio endpoint, che viene utilizzato dall'applicazione per le operazioni di lettura o scrittura sul nodo specifico. Oltre agli endpoint sul nodo primario, lo stesso cluster Memcached ha un endpoint denominato endpoint di configurazione. L'applicazione può utilizzare questo endpoint per leggere dal o scrivere sul cluster, lasciando la determinazione del nodo da cui leggere o scrivere all'individuazione automatica.



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

Configurazione dei cluster Valkey e Redis OSS

ElastiCache con Valkey e Redis OSS i cluster sono composti da 0 a 500 shard (chiamati anche gruppi di nodi). I dati in un cluster Valkey o Redis sono partizionati tra gli OSS shard del cluster. L'applicazione si connette a un OSS cluster Valkey o Redis utilizzando un indirizzo di rete chiamato Endpoint. I nodi di uno OSS shard Valkey o Redis svolgono uno dei due ruoli seguenti: uno primario di lettura/scrittura e tutti gli altri nodi secondari di sola lettura (chiamati anche repliche di lettura). Oltre agli endpoint del nodo, lo stesso cluster Valkey o Redis dispone di un endpoint chiamato endpoint di configurazione. OSS L'applicazione può utilizzare questo endpoint per leggere o scrivere nel cluster, lasciando la determinazione del nodo da cui leggere o su cui scrivere (Redis). ElastiCache OSS



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

ElastiCache requisiti di scalabilità

Tutti i cluster sono dimensionabili tramite la creazione di un nuovo cluster con un nuovo e più grande tipo di nodo. Quando si esegue la scalabilità di un cluster Memcached, il nuovo cluster inizia vuoto. Quando si esegue la scalabilità di un OSS cluster Valkey o Redis, è possibile eseguire il seeding da un backup ed evitare che il nuovo cluster sia inizialmente vuoto.

I cluster Amazon ElastiCache for Memcached possono essere scalati orizzontalmente o internamente. Per dimensionare un cluster Memcached basta aggiungere o rimuovere i nodi dal cluster. Se hai abilitato l'individuazione automatica e la tua applicazione si connette all'endpoint di configurazione del cluster, non devi apportare alcuna modifica all'applicazione quando aggiungi o rimuovi i nodi.

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

ElastiCache requisiti di accesso

In base alla progettazione, è possibile accedere ElastiCache ai cluster Amazon dalle EC2 istanze Amazon. L'accesso di rete a un ElastiCache cluster è limitato all'account che ha creato il cluster. Pertanto, prima di poter accedere a un cluster da un'EC2 istanza Amazon, devi autorizzare l'EC2 istanza Amazon ad accedere al cluster. I passaggi per eseguire questa operazione variano a seconda che tu abbia avviato in EC2 - VPC o EC2 -Classic.

Se hai avviato il cluster in EC2 - VPC devi concedere l'accesso alla rete al cluster. Se hai avviato il cluster in EC2 -Classic, devi concedere al gruppo di sicurezza Amazon Elastic Compute Cloud associato all'istanza l'accesso al tuo gruppo di ElastiCache sicurezza. Per istruzioni dettagliate, consultare [Fase 3. Autorizza l'accesso al cluster](#) in questa guida.

Requisiti per regione, zona di disponibilità e zona locale per ElastiCache

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 delle regioni e delle zone di disponibilità per ElastiCache](#)
- [Utilizzo delle zone locali con ElastiCache](#)
- [Limitazione dell'impatto degli errori](#)

Scelta delle dimensioni dei nodi

La dimensione del nodo selezionata per il ElastiCache cluster influisce sui costi, sulle prestazioni e sulla tolleranza agli errori.

Dimensione del nodo (Valkey e RedisOSS)

Per informazioni sui vantaggi dei processori Graviton, vedere [Processore AWS Graviton](#).

Rispondere alle seguenti domande può aiutarvi a determinare il tipo di nodo minimo necessario per l'implementazione di Valkey o Redis: OSS

- Si prevedono carichi di lavoro legati al velocità di trasmissione effettiva con più connessioni client?

Se questo è il caso e stai utilizzando la OSS versione Redis 5.0.6 o successiva, puoi ottenere velocità e latenza migliori con la nostra funzionalità di I/O avanzata, se disponibile, utilizzata per scaricare CPUs le connessioni client, per conto del motore Redis. OSS Se utilizzi la OSS versione Redis 7.0.4 o successiva, oltre all'I/O avanzato, otterrai un'accelerazione aggiuntiva con il multiplexing I/O avanzato, in cui ogni thread IO di rete dedicato trasferisce i comandi di più client nel OSS motore Redis, sfruttando la capacità di OSS Redis di elaborare in modo efficiente i comandi in batch. In ElastiCache (RedisOSS) v7.1 e versioni successive, abbiamo esteso la funzionalità avanzata dei thread di I/O per gestire anche la logica del livello di presentazione. Per livello di presentazione, ciò che intendiamo è che i thread di I/O avanzati ora non solo leggono l'input del client, ma lo analizzano anche nel formato di comando OSS binario Redis, che viene quindi inoltrato al thread principale per l'esecuzione, garantendo un aumento 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?

Se questo è il caso e utilizzi la versione 6.2 o successiva OSS del motore Redis, puoi sfruttare il tiering dei dati scegliendo il tipo di nodo r6gd. Con il tiering dei dati, i dati utilizzati meno di recente vengono archiviati in SSD. Quando vengono recuperati, è previsto un piccolo costo di latenza, bilanciato dai risparmi sui costi. Per ulteriori informazioni, consulta [Suddivisione dei dati su più livelli in ElastiCache](#).

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 OSS stai utilizzando?

OSSLe versioni di Redis precedenti alla 2.8.22 richiedono di riservare più memoria per il failover, le istantanee, la sincronizzazione e la promozione di una replica alle operazioni principali. Occorre infatti disporre di una quantità di memoria sufficiente per tutte le operazioni di scrittura necessarie, affinché il processo vada a buon fine.

La OSS versione Redis 2.8.22 e successive utilizzano un processo di salvataggio senza forkless che richiede meno memoria disponibile rispetto al processo precedente.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Modalità di implementazione di sincronizzazione e backup](#)
- [Assicurarsi di disporre di memoria sufficiente per creare un'istantanea Valkey o Redis OSS](#)
- 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. Ad esempio quando si scatta un'istantanea, si sincronizza un cluster primario con una replica in un cluster e quando si abilita la funzionalità append-only file (AOF). 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 [Assicurarsi di disporre di memoria sufficiente per creare un'istantanea Valkey o Redis OSS](#).

- La vostra implementazione sarà un cluster Valkey o Redis autonomo OSS (modalità cluster disabilitata) o un cluster Valkey o Redis OSS (modalità cluster abilitata) con più shard?

Cluster Valkey o Redis (modalità cluster disabilitata) OSS

Se stai implementando un cluster Valkey o Redis OSS (modalità cluster disabilitata), il tipo di nodo deve essere in grado di ospitare tutti i tuoi dati più l'overhead necessario, come descritto nel bullet 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.

Valkey o Redis OSS (modalità cluster abilitata) con più shard

Se stai implementando un cluster Valkey o Redis OSS (modalità cluster abilitata) con più shard, il tipo di nodo deve essere in grado di contenere 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?

[Le Local Zones](#) ti consentono di collocare risorse come un ElastiCache cluster in più posizioni vicine ai tuoi 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, è possibile monitorare l'utilizzo della memoria, l'utilizzo del processore, gli accessi alla cache e le metriche relative agli errori di 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 del nodo diversa con specifiche di memoria più grandi CPU.

Quando monitorate CPU l'utilizzo, ricordate che Valkey e Redis OSS sono a thread singolo. Pertanto, moltiplica l'CPU utilizzo riportato per il numero di CPU core per ottenere l'utilizzo effettivo. Ad esempio, un core a quattro core che CPU riporta un tasso di utilizzo del 20% è in realtà l'unico core che OSS Redis utilizza con un utilizzo dell'80%.

Dimensione del nodo (Memcached)

I cluster Memcached possono contenere uno o più nodi, tra i quali ripartire i propri dati. Per questo motivo, le esigenze di memoria del cluster e la memoria di un nodo sono correlate, ma non corrispondenti. La necessaria capacità di memoria del cluster può contare su pochi nodi grandi o molti nodi piccoli. Inoltre, in base alle mutevoli esigenze, è possibile aggiungere nodi al cluster o rimuoverli in qualunque momento, pagando solo ciò che occorre.

La capacità di memoria totale del cluster viene calcolata moltiplicando il numero di nodi del cluster per la RAM capacità di ciascun nodo al netto del sovraccarico del sistema. La capacità di un nodo dipende dalla sua tipologia.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

Il numero dei suoi nodi è un fattore determinante per la disponibilità di un cluster che esegue Memcached. Il fallimento di un solo nodo può influire sulla disponibilità dell'applicazione e sul carico sul database back-end. In tal caso, ElastiCache effettua il provisioning di un nodo sostitutivo per un nodo guasto e il nodo viene ripopolato. Per ridurre questo potenziale impatto sulla disponibilità, basta distribuire la memoria e la capacità di calcolo su più nodi a capacità ridotta, anziché su meno nodi ad ampia capacità.

Se, ad esempio, occorrono 35 GB di memoria cache, è possibile impostare una qualsiasi delle seguenti configurazioni:

- 11 nodi cache .t2.medium, ciascuno con 3,22 GB di memoria e 2 thread = 35,42 GB e 22 thread.
- 6 nodi cache .m4.large, ciascuno con 6,42 GB di memoria e 2 thread = 38,52 GB e 12 thread.
- 3 nodi cache .r4.large, ciascuno con 12,3 GB di memoria e 2 thread = 36,90 GB e 6 thread.

- 3 nodi cache .m4 .xlarge, ciascuno con 14,28 GB di memoria e 4 thread = 42,84 GB e 12 thread.

Opzioni disponibili per i nodi a confronto

Tipo di nodo	Memoria (in GB)	Core	Costo orario*	Nodi necessari	Memoria totale (in GiB)	Core totali	Costo mensile
cache.t2.medium	3,22	2	0,068 \$	11	35,42	22	538,56 \$
cache.m4.large	6,42	2	0,156 \$	6	38,52	12	673,92 \$
cache.m4.xlarge	14,28	4	0,311 \$	3	42,84	12	671,76 \$
cache.m5.xlarge	12,93	4	0,311 \$	3	38,81	12	671,76 \$
cache.m6g.large	6,85	2	\$0,147	6	41,1	12	\$635
cache.r4.large	12,3	2	0,228 \$	3	36,9	6	492,48 \$
cache.r5.large	13,07	2	0,216 \$	3	39,22	6	466,56 \$
cache.r6g.large	13,07	2	\$0,205	3	42,12	6	\$442

* Costo orario per nodo dall'8 ottobre 2020.

Costo mensile al 100% di utilizzo per 30 giorni (720 ore).

Ognuna di queste opzioni offre una disponibilità di memoria simile, ma capacità e costi di elaborazione diversi. Per confrontare i costi delle tue opzioni specifiche, consulta la pagina [ElastiCache dei prezzi di Amazon](#).

Nel caso dei cluster che eseguono Memcached, parte della memoria disponibile su ciascun nodo viene utilizzata per la gestione della connessione. Per ulteriori informazioni, consulta [Sovraccarico delle connessioni Memcached](#)

L'utilizzo di più nodi richiede la distribuzione delle chiavi tra gli stessi. Ogni nodo dispone del proprio endpoint. Per una gestione semplificata degli endpoint, puoi utilizzare ElastiCache la funzionalità Auto Discovery, che consente ai programmi client di identificare automaticamente tutti i nodi di un cluster. Per ulteriori informazioni, consulta [Identifica automaticamente i nodi del cluster \(Memcached\)](#).

In alcuni casi, è possibile non essere sicuri di quanta capacità è necessaria. In caso affermativo, per i test consigliamo di iniziare con un cache .m5.large nodo. Quindi monitora l'utilizzo della memoria, CPU l'utilizzo e la frequenza di accesso alla cache con le ElastiCache metriche pubblicate su Amazon. CloudWatch Per ulteriori informazioni sulle CloudWatch metriche per ElastiCache, consulta. [Monitoraggio dell'utilizzo con CloudWatch Metrics](#) Per la produzione e i carichi di lavoro più grandi, i nodi R5 offrono le migliori prestazioni e RAM il miglior rapporto qualità-prezzo.

Se il cluster non garantisce l'hit rate desiderato, basta aggiungere più nodi, aumentando così la memoria totale in esso disponibile.

Se il cluster è vincolato CPU ma ha una frequenza di successo sufficiente, configura un nuovo cluster con un tipo di nodo che fornisca maggiore potenza di calcolo.

Creazione di un cluster per Valkey o Redis OSS

Gli esempi seguenti mostrano come creare un OSS cluster Valkey o Redis utilizzando, and. AWS Management Console AWS CLI ElastiCache API

Creazione di un Valkey o Redis OSS (modalità cluster disabilitata) (Console)

ElastiCache supporta la replica quando si utilizza il motore Valkey o Redis. OSS Per monitorare la latenza tra il momento in cui i dati vengono scritti su un cluster primario di OSS lettura/scrittura Valkey o Redis e il momento in cui vengono propagati a un cluster secondario di sola lettura, aggiunge al cluster una chiave speciale, `ElastiCacheElastiCacheMasterReplicationTimestamp`. Questa chiave è l'ora corrente dell'Universal Time (UTC). Poiché un OSS cluster Valkey o Redis potrebbe essere aggiunto a un gruppo di replica in un secondo momento, questa chiave è inclusa in tutti i OSS cluster Valkey o Redis, anche se inizialmente non sono membri di un gruppo di replica. Per ulteriori informazioni sui gruppi di replica, consulta [Alta disponibilità utilizzando gruppi di replica](#).

Per creare un cluster Valkey o Redis OSS (modalità cluster disabilitata), procedi nel seguente modo. [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)

Non appena lo stato del cluster è disponibile, puoi concedere ad Amazon EC2 l'accesso, connetterti e iniziare a utilizzarlo. Per ulteriori informazioni, consulta [Fase 3. Autorizza l'accesso al cluster](#) e [Fase 4. Connect al 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 in ElastiCache](#).

Creazione di un cluster Valkey o Redis OSS (modalità cluster abilitata) (console)

Se utilizzi Redis OSS 3.2.4 o versione successiva, puoi creare un cluster Valkey o Redis OSS (modalità cluster abilitata). I cluster Valkey o Redis OSS (modalità cluster abilitata) supportano il partizionamento dei dati tra 1 e 500 shard (API/ gruppi di nodi) ma con alcune limitazioni. CLI Per un confronto tra Valkey o Redis OSS (modalità cluster disabilitata) e Valkey o Redis (modalità cluster abilitata), vedi. OSS [Motori e versioni supportati](#)

Per creare un cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando la console ElastiCache

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <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 i passaggi descritti in [Creazione di un cloud privato virtuale](#) (). VPC
5. Nella pagina della ElastiCache dashboard, scegli Crea cluster, quindi scegli Crea cluster Valkey o Crea cluster Redis OSS.
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 Cloud AWS , 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 \(RedisOSS\)](#) 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 Valkey e Redis OSS](#) e [Creazione di un gruppo di ElastiCache 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 [Suddivisione dei dati su più livelli in ElastiCache](#).

- e. Per Numero di shard, scegli il numero di shard (partizioni/gruppi di nodi) che desideri per questo cluster Valkey o Redis (modalità cluster abilitata). OSS

Per alcune versioni di Valkey o Redis OSS (modalità cluster abilitata), puoi modificare il numero di shard nel cluster in modo dinamico:

- Redis OSS 3.2.10 e versioni successive: se sul cluster è in esecuzione Redis OSS 3.2.10 o versioni successive, è possibile modificare il numero di shard nel cluster in modo dinamico. Per ulteriori informazioni, consulta [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#).
- Altre OSS versioni di Redis: se sul cluster è in esecuzione una versione di Redis precedente alla versione 3.2.10, esiste un altro approccio OSS. Per modificare il numero di partizioni nel cluster in questo caso, creare un nuovo cluster con il nuovo numero di 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 ogni partizione.

Esistono le seguenti restrizioni per Valkey o Redis OSS (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 ritieni di aver bisogno di più o meno repliche per shard (API/CLI: gruppo di nodi), devi creare un nuovo cluster con il nuovo numero di repliche. Per ulteriori informazioni, consulta [Tutorial: seminare 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 entrambi IPv6 gli indirizzi assegnati per funzionare in modalità dual-stack IPv4 e una sottorete -only per funzionare solo come -only. IPv6 IPv6

Quando crei un nuovo gruppo di sottoreti, inserisci l'ID a cui appartiene. VPC

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 in ElastiCache](#).
- [Crea una sottorete nel tuo VPC](#)

Se si esegue per [Utilizzo delle zone locali 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:

- 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 delle regioni e delle zone di disponibilità per ElastiCache](#).


5. Seleziona Successivo.
6. Nelle impostazioni Advanced Valkey o nelle impostazioni Advanced Redis OSS oppure
 - 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 gestita dal cliente e scegliendo la AWS KMS chiave. Per ulteriori informazioni, consulta [Utilizzo delle chiavi gestite dai clienti](#) di AWS KMS.

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#). Per Valkey 7.2 e versioni successive o il OSS motore Redis versione 6.0 e successive, se abiliti Encryption in-transit ti verrà 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 la console e CLI](#).

- AUTHUtente predefinito: meccanismo di autenticazione per un server Valkey o Redis. OSS Per ulteriori informazioni, vedere. [AUTH](#)
- AUTH— Un meccanismo di autenticazione per il server Valkey o RedisOSS. Per ulteriori informazioni, vedere. [AUTH](#)

 Note

Per OSS le versioni Redis comprese tra 3.2.6 e successive, esclusa la versione 3.2.10, 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. Puoi utilizzare il gruppo di sicurezza predefinito per il tuo VPC o crearne uno nuovo.

Per ulteriori informazioni sui gruppi di sicurezza, consulta [la sezione Gruppi di sicurezza per te VPC](#) nella Amazon VPC User Guide.

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, consulta [Snapshot e ripristino](#).

8. (Facoltativo) Specifica una finestra di manutenzione. La finestra di manutenzione è l'orario, generalmente di un'ora, ogni settimana in cui si ElastiCache pianifica la manutenzione del sistema per il cluster. È possibile ElastiCache scegliere il giorno e l'ora della finestra di manutenzione (nessuna preferenza) oppure scegliere autonomamente il giorno, l'ora e la durata (Specificare la finestra di manutenzione). Se dagli elenchi si sceglie Specify maintenance window (Specifica finestra di manutenzione), selezionare Start day (Giorno di inizio), Start time (Ora di inizio) e Duration (Durata) (in ore) per la finestra di manutenzione. Tutti gli orari sono UCT orari.

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

9. (Facoltativo) PerRegistro:

- In Formato registro, scegli Testo 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 [Taggare le tue risorse ElastiCache](#) .
 11. Scegli Next (Successivo).
 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 ElastiCache](#) (Redis) con Multi-AZ OSS
2. Segui la procedura descritta in [Utilizzo di outpost](#).

Per creare l'equivalente utilizzando ElastiCache API o al AWS CLI posto della ElastiCache console, consulta quanto segue:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Non appena lo stato del cluster è disponibile, puoi concedere EC2 l'accesso al cluster, connetterti e iniziare a usarlo. Per ulteriori informazioni, consulta [Fase 3. Autorizza l'accesso al cluster](#) e [Fase 4. Connect al 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 in ElastiCache](#).

Creazione di un cluster (AWS CLI)

Per creare un cluster utilizzando il 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 in ElastiCache](#).

Creazione di un cluster Valkey o Redis OSS (modalità cluster disabilitata) () CLI

Example — Un cluster Valkey o Redis OSS (modalità cluster disabilitata) senza repliche di lettura

Il CLI codice seguente crea un cluster di cache Valkey o Redis OSS (modalità cluster disabilitata) 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 Valkey o Redis OSS (modalità cluster abilitata) (AWS CLI)

I cluster Valkey o Redis OSS (modalità cluster abilitata) (API/CLI: gruppi di replica) non possono essere creati utilizzando l'operazione `create-cache-cluster`. Per creare un cluster Valkey o Redis OSS (modalità cluster abilitata) (API/CLI: gruppo di replica), vedere [Creazione da zero di un gruppo di replica Valkey o Redis OSS \(Cluster Mode Enabled\) \(AWS CLI\)](#).

Per ulteriori informazioni, consulta l'argomento AWS CLI di ElastiCache riferimento [create-replication-group](#).

Creazione di un cluster per Valkey o Redis OSS (ElastiCache API)

Per creare un cluster utilizzando il ElastiCache API, utilizzare l'operazione `CreateCacheCluster`.

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 in ElastiCache](#).

Argomenti

- [Creazione di un cluster di cache Valkey o Redis OSS \(modalità cluster disabilitata\) \(ElastiCache API\)](#)
- [Creazione di un cluster di cache in Valkey o Redis OSS \(modalità cluster abilitata\) \(ElastiCache API\)](#)

Creazione di un cluster di cache Valkey o Redis OSS (modalità cluster disabilitata) (ElastiCache API)

Il codice seguente crea un cluster di cache Valkey o Redis OSS (modalità cluster disabilitata) (ElastiCache API).

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 Valkey o Redis OSS (modalità cluster abilitata) () ElastiCache API

I cluster Valkey o Redis OSS (modalità cluster abilitata) (API/CLI: gruppi di replica) non possono essere creati utilizzando l'operazione. `CreateCacheCluster` Per creare un cluster Valkey o Redis OSS (modalità cluster abilitata) (API/CLI: gruppo di replica), vedere. [Creazione da zero di un gruppo di replica in Valkey o Redis OSS \(modalità cluster abilitata\) \(\) ElastiCache API](#)

Per ulteriori informazioni, consulta l'argomento di riferimento. ElastiCache API [CreateReplicationGroup](#)

Creazione di un cluster per Memcached

I seguenti esempi mostrano come creare un cluster utilizzando AWS Management Console, AWS CLI e. ElastiCache API

Creazione di un cluster Memcached (console)

Quando utilizzi il motore Memcached, Amazon ElastiCache supporta il partizionamento orizzontale dei dati su più nodi. Memcached permette l'individuazione automatica per non dover tenere traccia degli endpoint di ciascun nodo. Memcached monitora l'endpoint di ciascun nodo, aggiornando l'elenco degli endpoint man mano che i nodi vengono aggiunti e rimossi. Per interagire con il cluster, all'applicazione è sufficiente l'endpoint di configurazione.

Per creare un cluster Memcached, segui la procedura descritta in [Creazione di un cluster](#)

Non appena lo stato del cluster è disponibile, puoi concedere ad Amazon EC2 l'accesso, connetterti e iniziare a utilizzarlo. Per ulteriori informazioni, consulta [Fase 3. Autorizza l'accesso al cluster](#) e [Fase 4. Connect al 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 in ElastiCache](#).

Creazione di un cluster (AWS CLI)

Per creare un cluster utilizzando il 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 in ElastiCache](#).

Creazione di un cluster di cache Memcached (AWS CLI)

Il CLI codice seguente crea un cluster di cache Memcached con 3 nodi.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine memcached \  
--engine-version 1.4.24 \  
--cache-parameter-group default.memcached1.4 \  
--num-cache-nodes 3
```

Per Windows:

```
aws elasticache create-cache-cluster ^
--cache-cluster-id my-cluster ^
--cache-node-type cache.r4.large ^
--engine memcached ^
--engine-version 1.4.24 ^
--cache-parameter-group default.memcached1.4 ^
--num-cache-nodes 3
```

Creazione di un cluster per Memcached () ElastiCache API

Per creare un cluster utilizzando il ElastiCache API, utilizzare l'CreateCacheClusterazione.

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 in ElastiCache](#).

Argomenti

- [Creazione di un cluster di cache Memcached \(\) ElastiCache API](#)

Creazione di un cluster di cache Memcached () ElastiCache API

Il codice seguente crea un cluster Memcached con 3 nodi (). ElastiCache API

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
&Engine=memcached
&NumCacheNodes=3
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&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>
```

Visualizzazione dei dettagli di un ElastiCache cluster

È possibile visualizzare informazioni dettagliate su uno o più cluster utilizzando la ElastiCache console AWS CLI, oppure ElastiCache API.

Visualizzazione dei dettagli di un cluster Memcached (Console)

È possibile visualizzare i dettagli di un cluster Memcached utilizzando la ElastiCache console, il comando AWS CLI for ElastiCache o il. ElastiCache API

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

Per visualizzare i dettagli di un cluster Memcached

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione che ti interessa.
3. Nella dashboard del ElastiCache motore, scegli Memcached. In questo modo viene visualizzato un elenco di tutti i cluster in esecuzione sul motore Memcached.
4. Per visualizzare i dettagli di un cluster, scegliere la casella a sinistra del nome del cluster.
5. Per visualizzare informazioni sui nodi, scegli la scheda Nodes (Nodi), contenente le informazioni sullo stato dei nodi e sull'endpoint.
6. 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 CloudWatch Metrics](#)
7. Scegli la scheda Rete e sicurezza per visualizzare i dettagli sulla connettività di rete del cluster, sulla configurazione del gruppo di sottoreti e sul gruppo di sicurezza. VPC 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 dei ElastiCache cluster](#).
9. Scegli la scheda Tags (Tag) per visualizzare i dettagli su tutti i tag applicati alle risorse del cluster. Per ulteriori informazioni, consulta [Taggare le tue risorse ElastiCache](#).

Visualizzazione dei dettagli di Valkey o Redis OSS (modalità cluster disabilitata) (console)

È possibile visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster disabilitata) utilizzando la ElastiCache console, il AWS CLI for ElastiCache o il. ElastiCache API

La procedura seguente descrive in dettaglio come visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster disabilitata) utilizzando la console. ElastiCache

Per visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster disabilitata)

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nella dashboard del ElastiCache motore, scegli Valkey o Redis OSS per visualizzare un elenco di tutti i cluster in esecuzione su quel motore.
3. Per visualizzare i dettagli di un cluster, seleziona la casella di controllo a sinistra del nome del cluster. Assicurati di selezionare un cluster che esegue il OSS motore Valkey o Redis, non Clustered Valkey o Clustered Redis. OSS 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 CloudWatch Metrics](#)
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 dei ElastiCache cluster](#).

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 [Taggare le tue risorse ElastiCache](#).

Visualizzazione dei dettagli per un cluster Valkey o Redis OSS (Cluster Mode Enabled) (Console)

È possibile visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando la ElastiCache console, il AWS CLI for ElastiCache o il. ElastiCache API

La procedura seguente descrive come visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando la console. ElastiCache

Per visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster abilitata)

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione che ti interessa.
3. Nella dashboard del ElastiCache motore, scegli Valkey o Redis OSS per visualizzare un elenco di tutti i cluster in esecuzione su quel motore.
4. Per visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster abilitata), scegli la casella a sinistra del nome del cluster. Assicurati di scegliere un cluster che esegue il motore Valkey o Clustered Redis. OSS

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 CloudWatch Metrics](#)
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 Rete e sicurezza per visualizzare i dettagli sulla connettività di rete del cluster e sulla configurazione del gruppo di sottoreti, sul gruppo di VPC sicurezza e sull'eventuale metodo di crittografia 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 dei ElastiCache cluster](#).
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 [Taggare le tue risorse ElastiCache](#).

Visualizzazione dei dettagli di un ElastiCache cluster ()AWS CLI

Il codice seguente elenca i dettagli per *my-cluster*:

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

Replace (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 `create-cache-cluster` comando.

```
{
  "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 del cluster abilitato o disabilitato con 1 o più shard), usa il comando seguente per descrivere i dettagli del cluster (replace *my-cluster* con il nome del gruppo di replica (nome del cluster)):

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

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

Visualizzazione dei dettagli di un ElastiCache cluster (ElastiCache API)

È possibile visualizzare i dettagli di un cluster utilizzando l' ElastiCache APIDescribeCacheClustersazione. 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 ElastiCache API di riferimento [DescribeCacheClusters](#).

Modifica di un cluster ElastiCache

Oltre ad aggiungere o rimuovere nodi da un ElastiCache cluster, a volte è necessario apportare altre modifiche, 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 una particolare modifica ai parametri, consultate la sezione *Le modifiche hanno effetto nella colonna Dettagli nelle tabelle relative a [Parametri Memcached specifiche](#) [Parametri Valkey](#) e [Redis OSS](#)*. Per ulteriori informazioni sul riavvio dei nodi del cluster, consulta [Riavvio dei nodi](#).

Usando il ElastiCache AWS Management Console

Per modificare un cluster

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <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 OSS client Valkey o Redis di connettersi utilizzando sia la modalità cluster abilitata che la modalità cluster disabilitata. Dopo aver migrato tutti i OSS client Valkey o Redis per utilizzare la modalità cluster abilitata, è possibile completare la configurazione della modalità cluster e impostare la modalità cluster su Enabled.

- 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 [Gestione delle versioni per ElastiCache](#). Tuttavia, non puoi eseguire il downgrade alle versioni precedenti del motore se non eliminando il cluster esistente e creandolo di nuovo.

- VPCGruppi di sicurezza
- 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 SNS notifica

- Compatibilità della versione di Memcached Engine

- Tipo di rete

 Note

Se si passa da IPv4 a IPv6, è necessario selezionare o creare gruppi di sottoreti compatibili con IPv6. Per ulteriori informazioni, consulta [Scelta del tipo di rete in ElastiCache](#).

- VPC Gruppi di sicurezza
- Gruppo di parametri
- Finestra di manutenzione
- Argomento per la SNS notifica

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 apportate alle versioni del motore vengono applicate nella finestra di manutenzione successiva. Le altre modifiche, come quelle della finestra di manutenzione, vengono applicate immediatamente.

Per abilitare/disabilitare la consegna dei log per Redis

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 dei dettagli del cluster, scegli la scheda Registri.
3. Per abilitare o disabilitare gli slow log, scegli Abilita o Disabilita.

Se scegli l'abilitazione:

- a. In Formato registro, scegli Testo. JSON
- b. In Tipo di destinazione del registro, scegli CloudWatch Logs o Kinesis Firehose.
- c. In Destinazione del registro, puoi scegliere Crea nuovo e inserire il nome del gruppo di CloudWatchLogs log o il nome dello stream Kinesis Data Firehose. Puoi anche scegliere Seleziona esistente e quindi scegliere il nome del gruppo di CloudWatchLogs log o il nome dello stream Kinesis Data Firehose.

d. Scegli Abilita .

Per modificare la configurazione per Redis:


1. Scegli Modifica.
2. In Formato registro, scegli Testo. JSON
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.

Usare il con AWS CLI ElastiCache

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

È possibile eseguire l'aggiornamento alle versioni più recenti del motore Memcached. Per ulteriori informazioni in merito, consulta [Gestione delle versioni per ElastiCache](#). Tuttavia, non puoi eseguire il downgrade alle versioni precedenti del motore se non eliminando il cluster esistente e creandolo di nuovo.

 Important

È possibile eseguire l'aggiornamento alle versioni più recenti del motore Valkey o Redis. OSS Se si aggiornano le versioni principali del motore, ad esempio da Redis OSS 5.0.6 a Redis OSS 6.0, è necessario selezionare una famiglia di gruppi di parametri compatibile con la nuova versione del motore. Per ulteriori informazioni in merito, consulta [Gestione delle versioni per ElastiCache](#). 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, consulta l'argomento per AWS CLI ElastiCache [modify-cache-cluster](#)

Utilizzo del ElastiCache API

È possibile modificare un cluster esistente utilizzando l' `ElastiCache` `APIModifyCacheCluster` 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

È possibile eseguire l'aggiornamento alle versioni più recenti del motore Memcached. Per ulteriori informazioni in merito, consulta [Gestione delle versioni per ElastiCache](#). Tuttavia, non puoi eseguire il downgrade alle versioni precedenti del motore se non eliminando il cluster esistente e creandolo di nuovo.

Important

È possibile eseguire l'aggiornamento alle versioni più recenti del motore Valkey o Redis. OSS Se si aggiornano le versioni principali del motore, ad esempio da Redis OSS 5.0.6 a

Redis OSS 6.0, è necessario selezionare una famiglia di gruppi di parametri compatibile con la nuova versione del motore. Per ulteriori informazioni in merito, consulta [Gestione delle versioni per ElastiCache](#). 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, consultate l'argomento di riferimento. [ElastiCache API `ModifyCacheCluster`](#)

Aggiungere nodi a un ElastiCache cluster

L'aggiunta di nodi a un cluster Memcached aumenta il numero delle partizioni del cluster. Quando si modifica il numero delle partizioni in un cluster, alcuni dei keyspaces devono essere rimappati per essere mappati al nodo corretto. La rimappatura dei keyspaces aumenta temporaneamente il numero di mancati riscontri nella cache sul cluster. Per ulteriori informazioni, consulta [Configurazione ElastiCache del client per un bilanciamento efficiente del carico \(Memcached\)](#).

Per riconfigurare il cluster Valkey o Redis OSS (modalità cluster abilitata), vedi [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)

È possibile utilizzare la console di ElastiCache gestione AWS CLI o aggiungere nodi ElastiCache API al cluster.

Utilizzando il ElastiCache AWS Management Console

Se desideri aggiungere un nodo a un cluster Valkey o Redis OSS (modalità cluster disabilitata) a nodo singolo (uno senza replica abilitata), è necessario un processo in due fasi: prima aggiungi la replica e poi aggiungi un nodo di replica.

Argomenti

- [Per aggiungere la replica a un cluster Valkey o Redis senza shard OSS](#)
- [Per aggiungere nodi a un ElastiCache cluster \(console\)](#)

La procedura seguente aggiunge la replica a un Valkey o Redis a nodo singolo che non ha la replica abilitata. OSS 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 replica a un cluster Valkey o Redis senza shard OSS

1. Accedi a AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.

Viene visualizzato un elenco di cluster che eseguono quel motore.

3. Scegliere il nome di un cluster, non la casella alla sinistra del nome, a cui aggiungere i nodi.

Quanto segue è vero per un OSS cluster Redis in cui la replica non è abilitata:

- Sta eseguendo RedisOSS, non Clustered Redis. OSS
- 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 ElastiCache 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 Aggiungi.

Non appena lo stato del cluster ritorna disponibile, puoi passare alla procedura successiva e aggiungere repliche al cluster.

Per aggiungere nodi a un ElastiCache cluster (console)

La procedura seguente può essere utilizzata per aggiungere nodi a un cluster:

1. Accedi a AWS Management Console e apri la ElastiCache console 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 tuo cluster è un cluster Valkey o Redis OSS (modalità cluster abilitata), vedi. [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)

Se il tuo cluster è un cluster Valkey o Redis OSS (modalità cluster disabilitata) con zero shard, completa innanzitutto i passaggi indicati in. [Per aggiungere la replica a un cluster Valkey o Redis senza shard OSS](#)

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	Eliminazione	Eliminazione	<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	Eliminazione	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	Eliminazione	<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>Apply Immediately</code> - Sì, tutte le richieste di creazione vengono eseguite immediatamente. Se la nuova richiesta di creazione è impostata su <code>Apply Immediately</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.

Usare il con AWS CLI ElastiCache

Per aggiungere nodi a un cluster utilizzando il AWS CLI, utilizzare l' AWS CLI operazione `modify-cache-cluster` con i seguenti parametri:

- `--cache-cluster-id` L'ID del cluster di cache a cui aggiungere nodi.
- `--num-cache-nodes` Il parametro `--num-cache-nodes` specifica il numero di nodi nel cluster dopo l'applicazione della modifica. Per aggiungere nodi al cluster, `--num-cache-nodes` deve essere maggiore del numero attuale di nodi nel cluster. Se questo valore è inferiore al numero corrente di nodi, ElastiCache prevede la rimozione del parametro `cache-node-ids-to-remove` e di un elenco di nodi dal cluster. Per ulteriori informazioni, consulta [Usare il con AWS CLI ElastiCache](#).
- `--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 modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 5 \  
  --apply-immediately
```

Per Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 5 ^  
  --apply-immediately
```

Questa operazione produce un output simile al seguente (JSONformato):

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",  
    "PreferredAvailabilityZone": "us-west-2b",  
    "ConfigurationEndpoint": {  
      "Port": 11211,  
      "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"  
    }  
  }  
}
```

```

    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "NumCacheNodes": 2,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "SecurityGroups": [
      {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
      }
    ],
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {
      "NumCacheNodes": 5
    },
    "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
    "CacheNodeType": "cache.m3.medium",
    "DataTiering": "disabled",
  }
}

```

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

Usare il AWS CLI con ElastiCache

Se si desidera aggiungere nodi a un cluster Valkey o Redis esistente OSS (modalità cluster disabilitata) che non ha la replica abilitata, è necessario prima creare il gruppo di replica specificando il cluster esistente come primario. Per ulteriori informazioni, consulta [Creazione di un gruppo di replica utilizzando un cluster di cache Valkey o Redis OSS 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 utilizzando il AWS CLI, utilizzate l' AWS CLI operazione `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 produce un output simile al seguente (JSONformato):

```
{  
  "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' AWS CLI argomento [increase-replica-count](#).

Utilizzo del ElastiCache API

Se desideri aggiungere nodi a un cluster Valkey o Redis esistente OSS (modalità cluster disabilitata) che non ha la replica abilitata, devi prima creare il gruppo di replica specificando il cluster esistente come primario. Per ulteriori informazioni, consulta [Aggiungere repliche a un cluster Valkey o Redis OSS \(Cluster Mode Disabled\) autonomo \(\) ElastiCache API](#). Una volta che il gruppo di replica è disponibile, si può proseguire con il processo seguente.

Per aggiungere nodi a un cluster () ElastiCache API

- Chiamate l'IncreaseReplicaCountAPIoperazione con i seguenti parametri:
 - 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 questo valore è inferiore al numero corrente di nodi, usa il DecreaseReplicaCount API con il numero di nodi da rimuovere dal cluster.
 - ApplyImmediately Specifica se aggiungere i nodi subito o alla prossima finestra di manutenzione.
 - RegionSpecificare la AWS regione del cluster a cui si desidera 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, vedere l' ElastiCache API [l'operazione IncreaseReplicaCount](#).

Utilizzo del ElastiCache API

Per aggiungere nodi a un cluster (ElastiCache API)

- Chiamate l'operazione `ModifyCacheCluster` API con i seguenti parametri:
 - `CacheClusterId` L'ID del cluster a cui aggiungere nodi.
 - `NumCacheNodes` Il parametro `NumCacheNodes` specifica il numero di nodi nel cluster dopo l'applicazione della modifica. Per aggiungere nodi al cluster, `NumCacheNodes` deve essere maggiore del numero attuale di nodi nel cluster. Se questo valore è inferiore al numero corrente di nodi, ElastiCache si aspetta che il parametro `CacheNodeIdsToRemove` con un elenco di nodi venga rimosso dal cluster (vedi [Utilizzo di ElastiCache API con Memcached](#)).
 - `ApplyImmediately` Specifica se aggiungere i nodi subito o alla prossima finestra di manutenzione.
 - `Region` Specificare la AWS regione del cluster a cui si desidera aggiungere nodi.

L'esempio seguente mostra una chiamata per aggiungere nodi a un cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheCluster
&ApplyImmediately=true
```

```
&NumCacheNodes=5
&CacheClusterId=my-cluster
&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, vedere l' ElastiCache APIargomento [ModifyCacheCluster](#).

Rimozione di nodi da un ElastiCache cluster

È possibile eliminare un nodo da un cluster Valkey, Redis OSS o Memcached utilizzando il AWS Management Console, il o il AWS CLI. ElastiCache API

Note

Ogni volta che si cambia il numero di nodi in un cluster Memcached, è necessario rimappare almeno una parte del keyspace in modo da mapparlo sul nodo corretto. Per informazioni più dettagliate sul bilanciamento del carico di un cluster Memcached, consulta [Configurazione ElastiCache del client per un bilanciamento efficiente del carico \(Memcached\)](#):

Usando il ElastiCache AWS Management Console

Per rimuovere nodi da un cluster (console)

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione del cluster da cui desideri 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 ElastiCache console, puoi eliminare solo un nodo alla volta, quindi la scelta di più nodi significa che non puoi utilizzare il pulsante Elimina nodo.

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

Con Valkey o RedisOSS, se stai eliminando i risultati del nodo nel cluster che non sono più conformi a Multi-AZ, assicurati di deselezionare prima la casella di controllo Multi-AZ e poi di eliminare il nodo. Se si deseleziona 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	Eliminazione	Eliminazione	<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	Eliminazione	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	Eliminazione	<p>La nuova richiesta di eliminazione, in sospeso o immediata, sostituisce la richiesta di creazione in sospeso.</p>

Scenari	Operazioni in sospenso	Nuova richiesta	Risultati
			<p>Ad esempio, se è presente una richiesta in sospenso 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 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 data-bbox="727 877 1507 1335" 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 Applica immediatamente - Sì, tutte le richieste di creazione vengono eseguite immediatamente. Se la nuova richiesta di creazione è impostata su Applica immediatamente - 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.

Usare il con AWS CLI ElastiCache

1. `IDs`Identifica i nodi che desideri rimuovere. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di un ElastiCache cluster](#).
2. Utilizzate l'`decrease-replica-count`CLI operazione 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 nodi IDs che desideri rimuovere da questo cluster.
- `--apply-immediately` o `--no-apply-immediately` Specificano se rimuovere i nodi subito o alla prossima finestra di manutenzione.
- `--region` Specifica la AWS regione del cluster da cui si desidera 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 produce un output simile al seguente (JSON formato):

```
{  
  "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 AWS CLI argomenti [decrease-replica-count](#).

Utilizzo di ElastiCache API con Valkey o Redis OSS

Per rimuovere i nodi utilizzando il ElastiCache API, chiamate l'DecreaseReplicaCountAPIoperazione con l'ID del gruppo di replica e un elenco di nodi da rimuovere, come illustrato:

- `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 AWS regione del cluster da cui si desidera 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, vedere l' ElastiCache APIargomento [DecreaseReplicaCount](#).

Utilizzo di ElastiCache API con Memcached

Per rimuovere i nodi utilizzando il ElastiCache API, chiamate l'`ModifyCacheCluster` API operazione con l'ID del cluster di cache e un elenco di nodi da rimuovere, come mostrato:

- `CacheClusterId` L'ID del cluster di cache da cui rimuovere i nodi.
- `NumCacheNodes` Il parametro `NumCacheNodes` specifica il numero di nodi nel cluster dopo l'applicazione della modifica.
- `CacheNodeIdsToRemove.member.n` L'elenco dei nodi IDs da rimuovere dal cluster.
 - `CacheNodeIdsToRemove.member.1=0004`
 - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` Specifica se rimuovere i nodi subito o alla prossima finestra di manutenzione.
- `Region` Specifica la AWS regione del cluster da cui si desidera 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=ModifyCacheCluster  
  &CacheClusterId=my-cluster  
  &ApplyImmediately=true  
  &CacheNodeIdsToRemove.member.1=0004  
  &CacheNodeIdsToRemove.member.2=0005  
  &NumCacheNodes=3  
  &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, vedere l' [ElastiCache API argomento `ModifyCacheCluster`](#).

Annullamento delle operazioni di aggiunta o eliminazione dei nodi in sospeso in ElastiCache

Se si è scelto di non applicare immediatamente una modifica al ElastiCache cluster, l'operazione rimarrà in sospeso fino a quando non verrà eseguita nella finestra di manutenzione successiva. Si possono annullare tutte le operazioni in attesa.

Per annullare un'operazione in attesa

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione in cui desideri annullare un'operazione di aggiunta o eliminazione di un nodo in sospeso.
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 in ElastiCache

Finché un ElastiCache cluster è disponibile, ti verrà addebitato un costo, indipendentemente dal fatto che tu lo stia utilizzando attivamente o meno. Per interrompere l'addebito, elimina il cluster.

Warning

Quando si elimina un ElastiCache cluster, le istantanee manuali vengono conservate. È inoltre possibile creare uno snapshot finale prima che il cluster venga eliminato. Gli snapshot di cache automatici non vengono conservati.

Utilizzando il AWS Management Console

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

Per eliminare un cluster

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nella dashboard del ElastiCache motore, scegli il motore in cui è in esecuzione il cluster che desideri 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 solo un cluster alla volta dalla ElastiCache console. 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 per eliminare un ElastiCache cluster

Il codice seguente elimina il cluster `my-cluster` di ElastiCache cache.

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

L'operazione `delete-cache-cluster` elimina solo un 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, vedere l' [ElastiCache argomento AWS CLI `delete-cache-cluster`](#) per.

Utilizzo del ElastiCache API

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 `DeleteCacheCluster` elimina solo un 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 ElastiCache API [DeleteCacheCluster](#) di riferimento.

Accesso al ElastiCache cluster o al gruppo di replica

Le tue ElastiCache istanze Amazon sono progettate per essere accessibili tramite un'EC2istanza Amazon.

Se hai avviato l' ElastiCache istanza in un Amazon Virtual Private Cloud (AmazonVPC), puoi accedere all' ElastiCacheistanza da un'EC2istanza Amazon nella stessa AmazonVPC. Oppure, utilizzando il VPC peering, puoi accedere alla tua ElastiCache istanza da un Amazon EC2 in un altro AmazonVPC.

Se hai avviato l' ElastiCache istanza nella EC2 versione Classic, consenti all'EC2istanza di accedere al tuo cluster concedendo al gruppo di EC2 sicurezza Amazon associato all'istanza l'accesso al tuo gruppo di sicurezza della 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

Hai lanciato il tuo cluster in EC2 - VPC

Se hai avviato il cluster in un Amazon Virtual Private Cloud (AmazonVPC), puoi connetterti al ElastiCache cluster solo da un'EC2istanza Amazon in esecuzione nella stessa AmazonVPC. 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, VPC verrai esteso a quella zona locale e tratterai la sottorete come qualsiasi sottorete in qualsiasi altra zona di disponibilità e le considerazioni relative ai gateway, alle tabelle di routing e ad altre considerazioni relative ai gruppi di sicurezza verranno modificate automaticamente. VPC

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

1. Accedi a AWS Management Console e apri la EC2 console Amazon 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 tuo AmazonVPC. A meno che tu non abbia creato un gruppo di sicurezza da ElastiCache utilizzare, questo gruppo di sicurezza verrà denominato predefinito.
4. Scegli la scheda In entrata, quindi procedi come segue:
 - a. Scegli Modifica.
 - b. Scegli Aggiungi regola.
 - c. Nella colonna Tipo, scegli TCPRegola 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 Memcached è **11211** La porta predefinita per Valkey e Redis è. OSS **6379**
 - e. Nella casella Source, scegli Anywhere con l'intervallo di porte (0.0.0.0/0) in modo che qualsiasi EC2 istanza Amazon che avvii all'interno di Amazon VPC possa connettersi ai tuoi nodi. ElastiCache

 Important

L'apertura del ElastiCache cluster a 0.0.0.0/0 non espone il cluster a Internet perché non ha un indirizzo IP pubblico e quindi non è accessibile dall'esterno di VPC. Tuttavia, il gruppo di sicurezza predefinito può essere applicato ad altre EC2 istanze Amazon nell'account del cliente e tali istanze possono avere un indirizzo IP pubblico. Se eseguono qualche attività sulla porta predefinita, il servizio può essere esposto involontariamente. Pertanto, consigliamo di creare un gruppo VPC di sicurezza che verrà utilizzato esclusivamente da ElastiCache. Per ulteriori informazioni, consulta [Gruppi di sicurezza personalizzati](#).

- f. Seleziona Salva.

Quando avvii un'EC2istanza Amazon su AmazonVPC, quell'istanza sarà in grado di connettersi al tuo ElastiCache cluster.

Accesso alle ElastiCache risorse dall'esterno AWS

Amazon ElastiCache è un AWS servizio che fornisce un archivio chiave-valore in memoria basato sul cloud. Il servizio è progettato per essere accessibile esclusivamente dall'interno. AWS Tuttavia, se il ElastiCache cluster è ospitato all'interno di unVPC, è possibile utilizzare un'istanza Network Address Translation (NAT) per fornire l'accesso dall'esterno.

Requisiti

Per poter accedere alle ElastiCache risorse dall'esterno, è necessario soddisfare i seguenti requisiti AWS:

- Il cluster deve risiedere all'interno di un'istanza di Network Address Translation () VPC ed è accessibile tramite un'istanza di Network Address Translation (NAT). Non ci sono eccezioni a questo requisito.
- L'NATistanza deve essere avviata nello VPC stesso ambiente del cluster.
- L'NATistanza deve essere avviata in una sottorete pubblica separata dal cluster.
- Un indirizzo IP elastico (EIP) deve essere associato all'NATistanza. La funzionalità di port forwarding di iptables viene utilizzata per inoltrare una porta sull'NATistanza alla porta del nodo cache all'interno di. VPC

Considerazioni

Le seguenti considerazioni devono essere tenute a mente quando si accede alle risorse dall'esterno.

ElastiCache ElastiCache

- I client si connettono alla porta EIP e alla porta cache dell'NATistanza. Il port forwarding sull'NATistanza inoltra il traffico al nodo del 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'NATistanza funge da proxy tra 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 si accede tramite l'NATistanza.
- Il traffico dai client all'NATistanza non è crittografato. Pertanto, dovresti evitare di inviare dati sensibili tramite l'NATistanza.
- L'NATistanza aggiunge il sovraccarico derivante dalla manutenzione di un'altra istanza.
- L'NATistanza funge da singolo punto di errore. Per informazioni su come configurare l'alta disponibilità NAT suVPC, consulta [High Availability for Amazon VPC NAT Instances: An Example](#).

Come accedere alle ElastiCache risorse dall'esterno AWS

La procedura seguente mostra come connettersi alle ElastiCache risorse utilizzando un'NATistanza.

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`

Successivamente 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 Memcached con:
 - Indirizzo IP: 10.0.1.230
 - Porta Memcached predefinita: 11211
 - Gruppo di sicurezza: *10\0\0\0\55*
- Stai accedendo a un OSS cluster Valkey o Redis con:
 - Indirizzo IP: 10.0.1.230
 - Porta predefinita: 6379

- Gruppo di sicurezza: sg-bd56b7da
- AWS indirizzo IP dell'istanza: sg-bd56b7da
- L'indirizzo IP del client affidabile è 198.51.100.27.
- L'NATistanza ha l'indirizzo IP elastico 203.0.113.73.
- L'NATistanza ha il gruppo di sicurezza sg-ce56b7a9.

Per connetterti alle tue risorse usando un'istanza ElastiCache NAT

1. Crea un'NATistanza nello VPC stesso cluster di cache ma in una sottorete pubblica.

Per impostazione predefinita, la VPC procedura guidata avvia un tipo di nodo cache.m1.small. È necessario selezionare una dimensione di nodo in base alle proprie esigenze: È necessario utilizzare EC2 NAT AMI per poter accedere dall'esterno. ElastiCache AWS

Per informazioni sulla creazione di un'NATistanza, consulta [NATIstanze](#) nella Guida per l' AWS VPCutente.

2. Crea regole di gruppo di sicurezza per il cluster di cache e l'NATistanza.

Il gruppo di sicurezza dell'NATistanza e l'istanza del cluster devono avere le seguenti regole:

- Due regole in entrata
 - Con Memcached, la prima regola è consentire TCP le connessioni da client affidabili a ciascuna porta cache inoltrata dall'NATistanza (11211 - 11213).
 - Con Valkey e RedisOSS, la prima regola consiste nel consentire le TCP connessioni da client affidabili a ciascuna porta cache inoltrata dall'istanza (6379 - 6381). NAT
- Una seconda regola per consentire l'accesso a client affidabili. SSH

NATinstance security group: regole in entrata con Memcached

Tipo	Protocollo	Intervallo porte	Origine
Regola personalizzata TCP	TCP	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

NATgruppo di sicurezza dell'istanza: regole in entrata con Valkey o Redis OSS

Tipo	Protocollo	Intervallo porte	Origine
Regola personalizzata TCP	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Con Memcached, una regola in uscita per consentire TCP le connessioni alla porta cache (11211).

NATgruppo di sicurezza dell'istanza - regola in uscita

Tipo	Protocollo	Intervallo porte	Destinazione
Regola personalizzata TCP	TCP	11211	sg-ce56b7a9 (gruppo di sicurezza per le istanze del cluster)

- Con Valkey o RedisOSS, una regola in uscita per consentire TCP le connessioni alla porta cache (6379).

NATgruppo di sicurezza dell'istanza - regola in uscita

Tipo	Protocollo	Intervallo porte	Destinazione
Regola personalizzata TCP	TCP	6379	sg-ce56b7a9 (gruppo di sicurezza per le istanze del cluster)

- Con Memcached, una regola in entrata per il gruppo di sicurezza del cluster che consente TCP le connessioni dall'NATistanza alla porta cache (11211).

Gruppo di sicurezza per le istanze del cluster - Regola in entrata

Tipo	Protocollo	Intervallo porte	Origine
Regola personalizzata TCP	TCP	11211	sg-bd56b7da (Gruppo di sicurezza) NAT

- Con Valkey o RedisOSS, una regola in entrata per il gruppo di sicurezza del cluster che consente le TCP connessioni dall'istanza alla porta cache (6379). NAT

Gruppo di sicurezza per le istanze del cluster - Regola in entrata

Tipo	Protocollo	Intervallo porte	Origine
Regola personalizzata TCP	TCP	6379	sg-bd56b7da (gruppo di sicurezza del cluster)

3. Convalidare le regole.

- Verifica che il client affidabile sia in grado di SSH accedere all'NATistanza.
- Verifica che il client affidabile sia in grado di connettersi al cluster dall'NATistanza.

4. Memcached

Aggiungi una regola iptables all'istanza. NAT

È necessario aggiungere una regola iptables alla NAT tabella per ogni nodo del cluster per inoltrare la porta cache dall'NATistanza al nodo del cluster. Di seguito è riportato un possibile esempio:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to 10.0.1.230:11211
```

Il numero di porta deve essere univoco per ciascun nodo nel cluster. Ad esempio, se si utilizza un cluster Memcached a tre nodi tramite le porte 11211 - 11213, le regole avrebbero un aspetto simile al seguente:


```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
10.0.1.230:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to
10.0.1.231:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to
10.0.1.232:11211
```

Confermare che il client affidabile possa effettuare la connessione al cluster.

Il client affidabile deve connettersi all'istanza EIP associata all'NATistanza e alla porta del cluster corrispondente al nodo del cluster appropriato. Ad esempio, la stringa di connessione per PHP potrebbe essere simile alla seguente:

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

È anche possibile utilizzare un client telnet per verificare la connessione. Per esempio:

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

Valkey o Redis OSS

Aggiungi una regola iptables all'istanza. NAT

È necessario aggiungere una regola iptables alla NAT tabella per ogni nodo del cluster per inoltrare la porta cache dall'NATistanza 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. Ad esempio, se si lavora con un OSS cluster Redis a tre nodi utilizzando le porte 6379 - 6381, le regole sarebbero le seguenti:

```
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
```

Confermare che il client affidabile possa effettuare la connessione al cluster.

Il client affidabile deve connettersi all'istanza EIP associata all'NATistanza e alla porta del cluster corrispondente al nodo del cluster appropriato. Ad esempio, la stringa di connessione per PHP potrebbe essere simile alla seguente:

```
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
```

5. 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 per accedere a una ElastiCache cache in Amazon VPC](#)
- [Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente](#)
- [NATIstanze](#)
- [Configurazione dei client ElastiCache](#)

- [Alta disponibilità per le VPC NAT istanze Amazon: un esempio](#)

Ricerca degli endpoint di connessione in ElastiCache

L'applicazione si connette al ElastiCache cluster utilizzando gli endpoint. Un endpoint corrisponde all'indirizzo univoco di un nodo o di un cluster.

Quali endpoint utilizzare con Valkey o Redis OSS

- nodo autonomo, utilizza l'endpoint del nodo per operazioni di lettura e scrittura.
- I cluster Valkey o Valkey o Redis OSS (modalità cluster disabilitata) utilizzano l'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. Utilizza i singoli Node Endpoint per le operazioni di lettura (inAPI/CLI questi sono denominati Read Endpoints).
- I cluster Valkey o Redis OSS (modalità cluster abilitata), utilizzano l'endpoint di configurazione del cluster per tutte le operazioni che supportano i comandi abilitati in modalità cluster. È necessario utilizzare un client che supporti Valkey Cluster o Redis Cluster (Redis 3.2.OSS). OSS È ancora possibile leggere dagli endpoint dei singoli nodi (inAPI/CLI questi sono indicati come Read Endpoints).

Nelle sezioni seguenti è illustrato, fase per fase, il processo di individuazione degli endpoint necessari per il motore in esecuzione.

Quali endpoint usare con Memcached

Per una cache ElastiCache serverless con Memcached, è sufficiente acquisire l'endpoint DNS e la porta del cluster dalla console.

Da AWS CLI, usa il `describe-serverless-caches` comando per acquisire le informazioni sull'endpoint.

Linux

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

Windows

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

L'output dell'operazione precedente dovrebbe assomigliare a questo (JSONformato):

```
{
  "ServerlessCaches": [
    {
      "ServerlessCacheName": "serverless-memcached",
      "Description": "test",
      "CreateTime": 1697659642.136,
      "Status": "available",
      "Engine": "memcached",
      "MajorEngineVersion": "1.6",
      "FullEngineVersion": "21",
      "SecurityGroupIds": [
        "sg-083eda453e1e51310"
      ],
      "Endpoint": {
        "Address": "serverless-memcached-01.amazonaws.com",
        "Port": 11211
      },
      "ARN": "<the ARN>",
      "SubnetIds": [
        "subnet-0cf759df15bd4dc65",
        "subnet-09e1307e8f1560d17"
      ],
      "SnapshotRetentionLimit": 0,
      "DailySnapshotTime": "03:00"
    }
  ]
}
```

Per un'istanza basata su cluster Memcached, se usi l'individuazione automatica puoi utilizzare l'endpoint di configurazione del cluster per impostare il client Memcached. È necessario quindi utilizzare un client che supporti la funzione di individuazione automatica Auto Discovery.

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.

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

Se un cluster Valkey o Valkey o Redis OSS (modalità cluster disabilitata) ha un solo nodo, l'endpoint del nodo viene utilizzato sia per le letture che per le scritture. Se un cluster Valkey o Valkey o Redis OSS (modalità cluster disabilitata) ha più nodi, esistono tre tipi di endpoint: l'endpoint primario, l'endpoint reader e gli endpoint del nodo.

L'endpoint primario è un DNS nome che si risolve sempre nel nodo primario del 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 suddividerà in modo uniforme le connessioni in entrata all'endpoint tra tutte le repliche di lettura in un cluster (Redis). ElastiCache OSS 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. È possibile posizionare le repliche di lettura multiple del cluster ElastiCache (RedisOSS) in diverse zone di disponibilità (AZ) per garantire l'elevata AWS disponibilità degli endpoint di lettura.

Note

Un endpoint di lettura non è un servizio di load balancer. È un DNS record che verrà risolto 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 trovare gli endpoint di un cluster Valkey o Valkey o Redis OSS (modalità cluster disabilitata)

1. Accedi a AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli cluster Valkey o cluster Redis OSS.

Apparirà la schermata dei cluster con un elenco di cluster Valkey o Valkey o Redis (modalità cluster disabilitata) e Valkey o Redis OSS (modalità cluster abilitata). OSS

- 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	Description	Node type cache.r6g.large	Status Available
Engine Redis OSS	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 [lock icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [lock icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Endpoint primari e Reader per un cluster Valkery o Valkey o Redis (modalità cluster disabilitata) OSS

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

- Se il cluster Valkery o Valkey o Redis OSS (modalità cluster disabilitata) dispone di nodi di replica, è possibile trovare gli endpoint del nodo di replica del cluster scegliendo il nome del cluster e quindi selezionando la scheda 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 nodi per un cluster Valkery o Valkey o Redis (modalità cluster disabilitata) OSS

- 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, consulta [Connessione ai nodi Memcached](#).

Un endpoint primario Valkery o Valkey o Redis OSS (modalità cluster disabilitata) ha un aspetto simile al seguente. 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
```

Ricerca degli endpoint per un cluster (console) Valkey o Redis OSS (modalità cluster abilitata)

Un cluster Valkey o Redis OSS (abilitato in modalità cluster) ha un unico 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 Valkey o Redis OSS (modalità cluster abilitata)

1. Accedi a AWS Management Console e apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>

2. Dal pannello di navigazione, scegli cluster Valkey o cluster Redis OSS.

Apparirà la schermata dei cluster con un elenco di cluster. Scegli il cluster a cui desideri connetterti.

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.

Ricerca degli endpoint di un cluster (console) (Memcached)

Tutti gli endpoint Memcached sono sia di lettura che di scrittura. Per connettersi ai nodi in un cluster Memcached, l'applicazione può utilizzare gli endpoint di ciascun nodo o l'endpoint di configurazione del cluster, insieme all'individuazione automatica. Per avvalersi dell'individuazione automatica è necessario utilizzare un client che la supporti.

Con l'individuazione automatica, l'applicazione client si connette al cluster Memcached tramite l'endpoint di configurazione. Nel momento in cui si ridimensiona il cluster aggiungendo o rimuovendo nodi, l'applicazione viene automaticamente e immediatamente a "conoscenza" di tutti i nodi del cluster ed è in grado di connettersi a essi. Senza l'individuazione automatica, l'applicazione deve fare da sé o spetta all'utente aggiornare manualmente gli endpoint nell'applicazione, per ogni aggiunta o rimozione di un nodo.

Per copiare un endpoint, scegli l'icona di copia direttamente davanti all'indirizzo dell'endpoint. Per informazioni sull'utilizzo dell'endpoint per la connessione a un nodo, consulta [Connessione ai nodi Memcached](#).

Gli endpoint di configurazione e dei nodi sono molto simili. Le differenze sono riportate di seguito in grassetto.

```
myclustername.xxxxxx.cfg.usw2.cache.amazonaws.com:port # configuration endpoint  
contains "cfg"  
myclustername.xxxxxx.0001.usw2.cache.amazonaws.com:port # node endpoint for node 0001
```

Important

Se si sceglie di creare un CNAME endpoint di configurazione Memcached, affinché il client di rilevamento automatico lo riconosca CNAME come endpoint di configurazione, è necessario includere nel. .cfg. CNAME

Individuazione degli endpoint (AWS CLI)

Per Memcached, puoi usare for AWS CLI Amazon ElastiCache per scoprire gli endpoint per nodi e cluster.

Per Redis, puoi usare AWS CLI for Amazon per ElastiCache scoprire gli endpoint per nodi, cluster e anche gruppi di replica.

Argomenti

- [Individuazione degli endpoint per nodi e cluster \(AWS CLI\)](#)
- [Individuazione degli endpoint per i gruppi di OSS replica Valkey o Redis \(AWS CLI\)](#)

Individuazione degli endpoint per nodi e cluster (AWS CLI)

Puoi usare il AWS CLI per scoprire gli endpoint di un cluster e dei suoi nodi con il comando `describe-cache-clusters`. Per i cluster Valkey o Redis, il comando restituisce l'endpoint del OSS cluster. Per i cluster Memcached, il comando restituisce l'endpoint di configurazione. 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 l'endpoint di configurazione (`ConfigurationEndpoint`) e gli endpoint dei singoli nodi (`Endpoint`) del cluster Memcached `mycluster`.

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-clusters \
  --cache-cluster-id mycluster \
  --show-cache-node-info
```

Per Windows:

```
aws elasticache describe-cache-clusters ^
  --cache-cluster-id mycluster ^
  --show-cache-node-info
```

L'output dell'operazione precedente dovrebbe avere un aspetto simile al seguente (formato). JSON

```
{
```

```
"CacheClusters": [
{
  "Engine": "memcached",
  "CacheNodes": [
    {
      "CacheNodeId": "0001",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    },
    {
      "CacheNodeId": "0002",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    },
    {
      "CacheNodeId": "0003",
      "Endpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
      },
      "CacheNodeStatus": "available",
      "ParameterGroupStatus": "in-sync",
      "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
      "CustomerAvailabilityZone": "us-west-2b"
    }
  ],
  "CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "CacheParameterGroupName": "default.memcached1.4",
    "ParameterApplyStatus": "in-sync"
  },
  "CacheClusterId": "mycluster",
```

```

    "PreferredAvailabilityZone": "us-west-2b",
    "ConfigurationEndpoint": {
        "Port": 11211,
        "Address": "mycluster.amazonaws.com"
    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "available",
    "NumCacheNodes": 3,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
    "CacheNodeType": "cache.m4.large",
    "DataTiering": "disabled"
}
]
}

```

Important

Se scegli di creare un CNAME endpoint di configurazione Memcached, affinché il tuo client di auto discovery lo riconosca CNAME come endpoint di configurazione, devi includere nel `.cfg`. CNAME Ad esempio, `mycluster.cfg.local` nel file `php.ini` per il parametro `session.save_path`.

Example

Per Valkey e RedisOSS, il comando seguente recupera le informazioni sul cluster a nodo singolo `mycluster`.

Important

Il parametro `--cache-cluster-id` può essere utilizzato con ID cluster Valkey o Redis a nodo singolo OSS (modalità cluster disabilitata) o ID di nodo specifici nei gruppi di replica. Il valore `--cache-cluster-id` di un gruppo di replica è un valore a 4 cifre come `0001` Se

--cache-cluster-id è l'id di un cluster (nodo) in un gruppo di replica, replication-group-id viene 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 precedente dovrebbe avere un aspetto simile al seguente (JSONformato).

```
{  
  "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 i gruppi di OSS replica Valkey o Redis ()AWS CLI

È possibile utilizzare il AWS CLI per scoprire gli endpoint di un gruppo di replica e dei relativi cluster con il comando `describe-replication-groups`. 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'output di questa operazione dovrebbe essere simile al seguente (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, vedere [describe-replication-groups](#) nella Guida ai AWS CLI comandi.

Ricerca degli endpoint () ElastiCache API

Per Memcached, puoi usare Amazon ElastiCache API per scoprire gli endpoint per nodi e cluster.

Per Redis, puoi utilizzare Amazon ElastiCache API per scoprire gli endpoint per nodi, cluster e anche gruppi di replica.

Argomenti

- [Ricerca di endpoint per nodi e cluster \(\) ElastiCache API](#)
- [Ricerca degli endpoint per i gruppi di replica Valkey o Redis \(\) OSS ElastiCache API](#)

Ricerca di endpoint per nodi e cluster () ElastiCache API

Puoi usare il ElastiCache API per scoprire gli endpoint di un cluster e dei relativi nodi con l'azione. `DescribeCacheClusters` Per i cluster Valkey o Redis, il comando restituisce l'endpoint del OSS cluster. Per i cluster Memcached, il comando restituisce l'endpoint di configurazione. Se si include il parametro facoltativo `ShowCacheNodeInfo`, l'operazione restituisce, inoltre, gli endpoint dei vari nodi nel cluster.

Example

Per Memcached, il comando seguente recupera l'endpoint di configurazione (*ConfigurationEndpoint*) e gli endpoint dei singoli nodi () per il cluster Memcached `mycluster`.
Endpoint

```
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>
```

Important

Se scegli di creare un CNAME endpoint di configurazione Memcached, affinché il tuo client di auto discovery lo riconosca CNAME come endpoint di configurazione, devi includere

nel file `.cfg`. CNAME Ad esempio, `mycluster.cfg.local` nel file `php.ini` per il parametro `session.save_path`.

Ricerca degli endpoint per i gruppi di replica Valkey o Redis () OSS ElastiCache API

È possibile utilizzare l'azione ElastiCache API per scoprire gli endpoint di un gruppo di replica e dei relativi cluster. `DescribeReplicationGroups` 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 del lettore () e gli endpoint dei singoli nodi (`ReaderEndpoint`) per il gruppo di replica. `ReadEndpoint myreplgroup` 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, vedere. [DescribeReplicationGroups](#)

Lavorare con i frammenti in ElastiCache

Uno shard (API/CLI: `node group`) è una raccolta da uno a sei ElastiCache con nodi Valkey o Redis. OSS Un cluster Valkey o Redis OSS (modalità cluster disabilitata) non avrà mai più di uno 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 di nodi o shard può essere aumentato fino a un massimo di 500 per cluster se la versione del motore è Valkey 7.2 o Redis 5.0.6 o superiore. OSS 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 insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo troppo piccolo o che le sottoreti sono condivise e utilizzate intensamente da altri cluster. Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

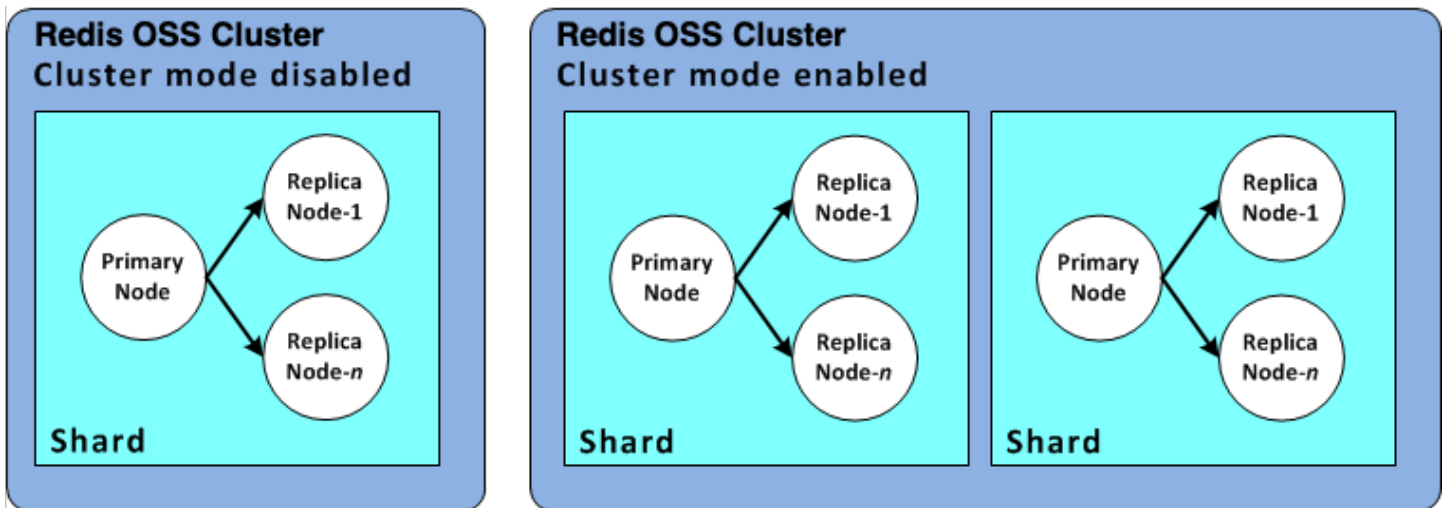
Per le versioni inferiori alla 5.0.6, il limite è 250 per cluster.

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 Valkey o Redis OSS (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 Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#). Se si utilizza ElastiCache API o AWS CLI per creare un cluster (chiamato gruppo di replica in API/CLI), è possibile 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. 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 di shard Valkey o Redis OSS

Per ulteriori informazioni, consulta [Resharding offline per Valkey o Redis \(modalità cluster abilitata\) OSS](#) e [Resharding online per Valkey o Redis OSS \(modalità cluster abilitata\)](#).

Ricerca dell'ID di uno shard

Puoi trovare l'ID di uno shard usando il, il o il AWS Management Console. AWS CLI ElastiCache API

Utilizzando il AWS Management Console

Argomenti

- [Per Valkey o Redis OSS \(modalità cluster disabilitata\)](#)
- [Per Valkey o Redis OSS \(modalità cluster abilitata\)](#)

Per Valkey o Redis OSS (modalità cluster disabilitata)

Gli shard del gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) sono sempre disponibili. IDs 0001

Per Valkey o Redis OSS (modalità cluster abilitata)

La procedura seguente utilizza il AWS Management Console per trovare l'ID shard del gruppo di replica di Valkey o Redis OSS (modalità cluster abilitata).

Per trovare l'ID dello shard in un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata)

1. Accedi a AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Nel pannello di navigazione, scegli Valkey o Redis OSS, quindi scegli il nome del gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) per cui desideri trovare lo shard. IDs
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 degli shard (gruppo di nodi) per i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata) o Valkey o Redis OSS (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 degli shard (gruppo di nodi) sono *highlighted* qui per facilitarne 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": {}
  }
]
```

Usando il ElastiCache API

Per trovare gli ID degli shard (gruppo di nodi) per i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata) o Valkey o Redis OSS (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 questo parametro viene omissso, i dettagli di un massimo di `xxx` vengono restituiti i 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 le cache autoprogettate da ValkeyOSS, Redis e Memcached

Amazon ElastiCache supporta i motori di cache Valkey, Redis OSS e Memcached. 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 una cache, un cluster o un gruppo di replica progettato autonomamente, puoi eseguire l'aggiornamento a una versione più recente del motore, ma non puoi effettuare il downgrade a una versione precedente del motore. Se si desidera utilizzare una versione precedente del motore, è necessario eliminare la cache, il cluster autoprogettato o il gruppo di replica esistente e crearlo nuovamente con la versione precedente del motore.

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.

Scegliete Valkey o Redis OSS con ElastiCache se vi si applica quanto segue:

- ElastiCache con Valkey 7.2 o Redis versione 7.0 (EnhancedOSS)

[Vuoi usare Functions, Sharded Pub/Sub o miglioramenti. ACL](#) Per ulteriori informazioni, consulta la [OSSversione Redis 7.0 \(Enhanced\)](#).

- ElastiCache (RedisOSS) versione 6.2 (Enhanced)

Desideri la possibilità di suddividere i dati tra la memoria e l'SSDutilizzo del tipo di nodo r6gd. Per ulteriori informazioni, consulta [Tiering di dati](#).

- ElastiCache (RedisOSS) versione 6.0 (avanzata)

Vuoi autenticare gli utenti con il controllo accessi basato sui ruoli.

Per ulteriori informazioni, consulta [Redis OSS versione 6.0 \(Enhanced\)](#).

- ElastiCache (RedisOSS) versione 5.0.0 (Enhanced)

Desideri utilizzare [Redis OSS Streams](#), una struttura di dati di registro che consente ai produttori di aggiungere nuovi articoli in tempo reale e consente inoltre ai consumatori di utilizzare i messaggi in modo bloccante o non bloccante.

Per ulteriori informazioni, consulta la [OSSversione Redis 5.0.0 \(Enhanced\)](#).

- ElastiCache (RedisOSS) versione 4.0.10 (Enhanced)


Supporta sia la crittografia che l'aggiunta o la rimozione dinamica di shard dal cluster Valkey o Redis OSS (modalità cluster abilitata).

Per ulteriori informazioni, consulta la [OSSversione 4.0.10 di Redis \(Enhanced\)](#).

Le seguenti versioni sono obsolete, hanno raggiunto o stanno per giungere a fine vita.

- ElastiCache (RedisOSS) versione 3.2.10 (Enhanced)

Supporta la possibilità di aggiungere o rimuovere dinamicamente shard dal cluster Valkey o Redis OSS (modalità cluster abilitata).

 Important

Attualmente ElastiCache (RedisOSS) 3.2.10 non supporta la crittografia.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [OSSVersione Redis 3.2.10 \(avanzata\)](#)
- Best practice di resharding online per Redis. Per ulteriori informazioni OSS, consulta quanto segue:
 - [Best practice: resharding online](#)
 - [Resharding online e ribilanciamento degli shard per Valkey o Redis \(modalità cluster abilitata\)](#)

- [Per ulteriori informazioni sulla scalabilità dei cluster Redis, consulta Scaling. OSS](#)

- ElastiCache (Redis) versione 3.2.6 (OSSEnhanced)

Se hai bisogno delle funzionalità delle OSS versioni precedenti di Redis oltre alle seguenti funzionalità, scegli ElastiCache (OSSRedis) 3.2.6:

- Crittografia dei dati in transito Per ulteriori informazioni, consulta [Amazon ElastiCache \(RedisOSS\) In-Transit Encryption](#).
- Crittografia dei dati inattivi Per ulteriori informazioni, consulta [Amazon ElastiCache \(RedisOSS\) At-Rest Encryption](#).
- ElastiCache (RedisOSS) (modalità cluster abilitata) versione 3.2.4

Se hai bisogno della funzionalità di Redis OSS 2.8.x oltre alle seguenti funzionalità, scegli Redis OSS 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 (RedisOSS) (modalità non cluster) 2.8.x e 3.2.4 (Enhanced)

Se si applica quanto segue, scegli Redis OSS 2.8.x o Redis 3.2.4 (modalità non in cluster): OSS

- 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 funzionalità di backup e ripristino per cluster progettati autonomamente e cache serverless.
- È necessario il support di più database.

Riepilogo del confronto tra Memcached, Valkey o Redis OSS (modalità cluster disabilitata) e Valkey o Redis (modalità cluster abilitata) OSS

	Memcached	Valkey o Redis (modalità cluster disabilitata) OSS	Valkey o Redis OSS (modalità cluster abilitata)
Versioni del motore+	1.4.5 e versioni successive	versioni 4.0.10 e successive	versioni 4.0.10 e successive
Tipi di dati	Semplici	2.8.x – Complessi * Complessi	3.2.x e versioni successive – Complessi
Partizionamento dei dati	Sì	No	Sì
Il cluster è modificabile	Sì	Sì	3.2.10 e versioni successive – Limitato
Resharding online	No	No	versioni 3.2.10 e successive
Crittografia	in-transit 1.6.12 e versioni successive	versioni 4.0.10 e successive	versioni 4.0.10 e successive
Tiering di dati	No	6.2 e versioni successive	6.2 e versioni successive
Certificazioni di conformità			
Certificazione di conformità			
Fed RAMP	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

	Memcached	Valkey o Redis (modalità cluster disabilitata) OSS	Valkey o Redis OSS (modalità cluster abilitata)
A più thread	Sì	No	No
Aggiornare il tipo di nodo	No	Sì	Sì
Aggiornamenti del motore	Sì	Sì	Sì
Elevata disponibilità (replica)	No	Sì	Sì
Failover automatico	No	Facoltativo	Richiesto
Capacità Pub/Sub	No	Sì	Sì
Serie ordinate	No	Sì	Sì
Backup e ripristino	Solo per Memcached Serverless, non per cluster Memcached progettati autonomam ente	Sì	Sì
Indicizzazione geospaziale	No	versioni 4.0.10 e successive	Sì

Note:

stringa, oggetti (database mi piace)

* stringa, set, set ordinati, elenchi, hash, bitmap, hyperloglog

stringa, set, set ordinati, elenchi, hash, bitmap, hyperloglog, indici geospaziali

+ Esclude le versioni obsolete, che hanno raggiunto o stanno per scadere.

Dopo aver scelto il motore per il cluster, ti consigliamo di utilizzare la versione più recente per quel motore. Per ulteriori informazioni, consulta [Tipi di nodi supportati](#).

Migrazione online per Valkey o Redis OSS

Utilizzando Online Migration, puoi migrare i tuoi dati da Valkey o Redis open source ospitati autonomamente su Amazon ad AmazonOSS. EC2 ElastiCache

Note

La migrazione online non è supportata verso cache o cluster ElastiCache serverless in esecuzione sul tipo di nodo r6gd.

Panoramica

La migrazione dei dati da Valkey o Redis open source in esecuzione OSS su Amazon ad Amazon ElastiCache richiede una EC2 distribuzione Amazon esistente o di nuova creazione. ElastiCache 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 la migrazione dei dati da Valkey o Redis open source ospitati autonomamente OSS su EC2 Amazon ElastiCache e non per lo spostamento di dati tra cluster. ElastiCache

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 o il comando. `StartMigration` API AWS CLI Durante la migrazione di cluster Valkey o Redis disabilitati in OSS modalità cluster, il processo di migrazione rende il nodo primario del OSS cluster Valkey o Redis una replica del server principale ElastiCache Valkey o Redis di origine. OSS Durante la migrazione di cluster Valkey o Redis abilitati alla OSS modalità cluster, il processo di migrazione trasforma il nodo primario di ogni shard in una replica dello shard corrispondente del cluster di origine che possiede gli stessi slot. ElastiCache

Dopo che le modifiche sul lato client sono pronte, richiama l'operazione. CompleteMigration API Questa API operazione promuove l' ElastiCache implementazione verso la distribuzione principale di Valkey o Redis OSS con nodi primari e di replica (a seconda dei casi). Ora puoi reindirizzare l'applicazione client su cui iniziare a scrivere dati. ElastiCache Durante la migrazione, è possibile controllare lo stato della replica eseguendo il INFO comando [valkey-cli sui nodi Valkey](#) e sui nodi primari. ElastiCache

Fasi della migrazione

I seguenti argomenti delineano il processo di migrazione dei dati:

- [Preparazione dell'origine e della 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 dell'origine e della destinazione per la migrazione

Con questi passaggi puoi prepararti a migrare i tuoi dati da una fonte Valkey o Redis ospitata autonomamente verso o da un cluster Redis EC2 a ElastiCache un OSS cluster Valkey. ElastiCache

È necessario assicurarsi che tutti e quattro i prerequisiti indicati di seguito siano soddisfatti prima di iniziare la migrazione dalla console, oppure. ElastiCache API AWS CLI

Per preparare i nodi Valkey o Redis di origine e di destinazione per la migrazione OSS

1. Identifica l' ElastiCache implementazione di destinazione e assicurati di poter migrare i dati su di essa.

Una ElastiCache distribuzione esistente o appena creata deve soddisfare i seguenti requisiti per la migrazione:

- Utilizza Valkey o la versione 5.0.6 o successiva OSS del motore Redis.
- Non è abilitata la crittografia in transito.
- Ha la funzione Multi-AZ abilitata.

- Dispone di memoria sufficiente per contenere i dati del cluster Valkey o RedisOSS. Per configurare le impostazioni di memoria riservata corrette, consulta [Gestione della memoria riservata per Valkey e Redis OSS](#).
 - Se la modalità cluster è disattivata, è possibile migrare direttamente dalle OSS versioni 2.8.21 di Valkey o Redis in poi a Valkey o Redis dalla OSS versione 5.0.6 in poi se si utilizza la o Valkey o le versioni Redis 5.0.6 in poi utilizzando la console or. CLI OSS CLI Se la modalità cluster è abilitata, puoi migrare direttamente da qualsiasi versione Valkey o Redis abilitata in modalità cluster alla OSS OSS versione Redis 5.0.6 in poi, se utilizzi la CLI o OSS le versioni Redis 5.0.6 in poi utilizzando la console CLI or.
 - 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 del tuo Valkey o Redis open source e la distribuzione siano compatibili. OSS ElastiCache

Come minimo, tutte le seguenti funzionalità della ElastiCache distribuzione di destinazione devono essere compatibili con la configurazione Valkey o Redis per la replica: OSS

- Il tuo cluster non avrebbe dovuto essere abilitato. AUTH
 - La configurazione `protected-mode` deve essere impostata su. no
 - Se hai una `bind` configurazione nella configurazione di Valkey o RedisOSS, allora dovrebbe essere aggiornata per consentire le richieste dai nodi. ElastiCache
 - Il numero di database logici deve essere lo stesso sul ElastiCache nodo e sul cluster Valkey o Redis. OSS Questo valore viene impostato utilizzando `databases` la configurazione Valkey o Redis. OSS
 - OSS I comandi Valkey o Redis che eseguono la modifica dei dati non devono essere rinominati per consentire la corretta replica dei dati. ad esempio,,,, e. `sync` `psync` `info` `config` `command` `cluster`
 - Per replicare i dati dal OSS cluster Valkey o Redis ElastiCache, assicurati che ci sia memoria sufficiente per gestire questo carico aggiuntivo. CPU Questo carico proviene dal RDB file creato dal OSS cluster Valkey o Redis e trasferito tramite la rete al nodo. ElastiCache
 - Tutte le OSS istanze Valkey o Redis nel cluster di origine devono essere in esecuzione sulla stessa porta.
3. Assicurati che le tue istanze possano connettersi ElastiCache effettuando le seguenti operazioni:

- Verifica che l'indirizzo IP di ogni istanza sia privato.
 - Assegna o crea la ElastiCache distribuzione nello stesso cloud privato virtuale (VPC) di Valkey o Redis OSS sull'istanza (consigliato).
 - Se VPCs sono diversi, configura il VPC peering per consentire l'accesso tra i nodi. Per ulteriori informazioni sul VPC peering, consulta [Modelli di accesso per accedere a una ElastiCache cache in Amazon VPC](#)
 - Il gruppo di sicurezza collegato alle OSS istanze Valkey o Redis dovrebbe consentire il traffico in entrata dai nodi. ElastiCache
4. Assicurati che l'applicazione possa indirizzare il traffico verso ElastiCache i nodi dopo il completamento della migrazione dei dati. Per ulteriori informazioni, consulta [Modelli di accesso per accedere a una ElastiCache cache in Amazon VPC](#).

Test della migrazione dei dati

Una volta completati tutti i prerequisiti, è possibile convalidare la configurazione della migrazione utilizzando AWS Management Console ElastiCache API, o. AWS CLI L'esempio seguente mostra l'utilizzo di. 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.

Di seguito è riportato un esempio di utilizzo di 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

Una volta completati tutti i prerequisiti, è possibile iniziare la migrazione dei dati utilizzando AWS Management Console ElastiCache API, o 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 l'utilizzo di CLI.

Note

Si consiglia di utilizzarlo per TestMigration API 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`— Identificatore del gruppo di replica di destinazione ElastiCache
- `--customer-node-endpoint-list`— Un elenco di endpoint con uno DNS o più indirizzi IP e la porta su cui è in esecuzione il cluster Valkey o Redis OSS di origine. L'elenco può accettare solo un elemento sia per la modalità cluster disabilitata che per la modalità cluster abilitata. Se hai abilitato la replica a catena, l'endpoint può puntare a una replica anziché al nodo principale del cluster Valkey o Redis. OSS

Di seguito è riportato un esempio di utilizzo di CLI

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

Durante l'esecuzione di questo comando, il nodo ElastiCache primario (in ogni shard) si configura per diventare una replica dell'OSSistanza Valkey o Redis (nello shard corrispondente che possiede gli stessi slot nei redis abilitati al cluster). Lo stato del ElastiCache cluster cambia in fase di migrazione e i dati iniziano a migrare dall'istanza Valkey o Redis al nodo primario. OSS ElastiCache A seconda della dimensione dei dati e del carico sull'OSSistanza Valkey o Redis, il completamento della migrazione può richiedere del tempo. Puoi controllare l'avanzamento della migrazione eseguendo il INFO comando [valkey-cli sull'istanza Valkey e sul nodo](#) primario. ElastiCache

Dopo una replica riuscita, tutte le scritture sulle istanze Valkey o Redis si propagano al cluster. OSS ElastiCache È possibile utilizzare i nodi per le letture. ElastiCache Tuttavia, non è possibile scrivere nel ElastiCache cluster. Se a un nodo ElastiCache primario sono collegati altri nodi di replica, questi nodi di replica continuano a replicarsi dal nodo primario ElastiCache . In questo modo, tutti i dati del cluster Valkey o Redis vengono replicati su tutti i nodi del OSS cluster. ElastiCache

Se un nodo ElastiCache primario non riesce a diventare una replica dell'OSSistanza Valkey o Redis, riprova più volte prima di tornare infine al livello primario. Lo stato del ElastiCache cluster diventa quindi disponibile e viene inviato un evento del gruppo di replica relativo all'impossibilità di avviare la 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 dell'origine e della destinazione per la migrazione](#).
- Assicurati che la configurazione di routing per le tue sottoreti VPC e per le tue sottoreti consenta il traffico tra i ElastiCache nodi e le tue istanze Valkey o Redis. OSS
- Assicurati che il gruppo di sicurezza collegato alle tue istanze Valkey o Redis OSS consenta il traffico vincolato all'input proveniente dai nodi. ElastiCache
- Controlla i OSS log di Valkey o Redis per le tue istanze per ulteriori informazioni sugli 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:

- Verifica che Valkey o OSS `master_link_status` Redis sia up presente nel INFO comando sui nodi ElastiCache primari. Puoi trovare queste informazioni anche nella ElastiCache console. Seleziona il cluster e, nella sezione CloudWatch Metrics, osserva Primary Link Health Status. Una volta che il valore raggiunge 1, i dati sono sincronizzati.
- Puoi verificare che la ElastiCache replica abbia uno stato online eseguendo il INFO comando sulle tue istanze Valkey o Redis. OSS In questo modo è possibile ottenere informazioni sul rallentamento della replica.
- Verifica che il buffer di output del client sia insufficiente utilizzando il [CLIENTLIST](#) comando sulle istanze Valkey o Redis. OSS

Una volta completata la migrazione dei dati, i dati sono sincronizzati con tutte le nuove scritture in arrivo sui nodi primari del cluster Valkey o Redis. OSS

Completamento della migrazione dei dati

Quando sei pronto per passare al ElastiCache cluster, usa il `complete-migration` CLI comando 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 esegui questo comando, il nodo ElastiCache primario (in ogni shard) interrompe la replica dall'OSSistanza Valkey o Redis e lo promuove a primario. 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 alla scrittura e alla ElastiCache lettura. ElastiCache lo stato del cluster dovrebbe cambiare da migrante a disponibile.

Se la promozione al livello primario fallisce, il nodo ElastiCache primario continua a replicarsi dall'istanza Valkey o Redis. OSS Il ElastiCache cluster continua a essere in stato di migrazione e viene inviato un messaggio relativo all'errore del gruppo di replica. Per risolvere questo errore, seguire questa procedura:

- Controllare l'evento del gruppo di replica. Utilizzare le informazioni specifica dell'evento per correggere l'errore.
- Potrebbe essere generato un messaggio di evento sui dati non sincronizzati. In tal caso, assicurati che l'istanza ElastiCache primaria possa essere replicata dalla tua OSS istanza Valkey o Redis e che entrambe siano sincronizzate. Se si desidera ancora arrestare la migrazione, è possibile eseguire il comando precedente con l'opzione `--force`.
- Potresti ricevere un messaggio relativo all'evento se uno dei ElastiCache nodi 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

È possibile utilizzare il AWS Management Console per migrare i dati dal cluster al cluster Valkey o RedisOSS.

Per eseguire la migrazione dei dati online utilizzando la console

1. Accedi alla console e apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Crea un nuovo cluster Valkey o Redis o scegli un OSS cluster esistente. Verificare che il cluster soddisfi i seguenti requisiti:
 - La versione del motore deve essere Valkey 7.2 e versioni successive oppure Redis OSS 5.0.6 o versioni successive.
 - Il tuo cluster non avrebbe dovuto essere abilitato. AUTH
 - La configurazione `protected-mode` deve essere impostata su. `no`
 - Se hai una `bind` configurazione nella configurazione di Valkey o RedisOSS, allora dovrebbe essere aggiornata per consentire le richieste dai nodi. ElastiCache
 - Il numero di database deve essere lo stesso tra il ElastiCache nodo e il cluster Valkey o Redis. OSS Questo valore viene impostato utilizzando la configurazione `databases` del motore.
 - OSS I comandi Valkey o Redis che eseguono la modifica dei dati non devono essere rinominati per consentire la corretta replica dei dati.
 - Per replicare i dati dal OSS cluster Valkey o Redis ElastiCache, assicurati che ci sia memoria sufficiente CPU per gestire questo carico aggiuntivo. Questo carico proviene dal RDB file creato dal OSS cluster Valkey o 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 finestra di dialogo `Migrate Data from Endpoint`, inserisci l'indirizzo IP e la porta in cui è disponibile il cluster Valkey o Redis. OSS

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 delle regioni e delle zone di disponibilità per ElastiCache

È possibile fornire ulteriore scalabilità e affidabilità ai ElastiCache cluster designando regioni e zone di disponibilità utilizzando l'endpoint corrispondente.

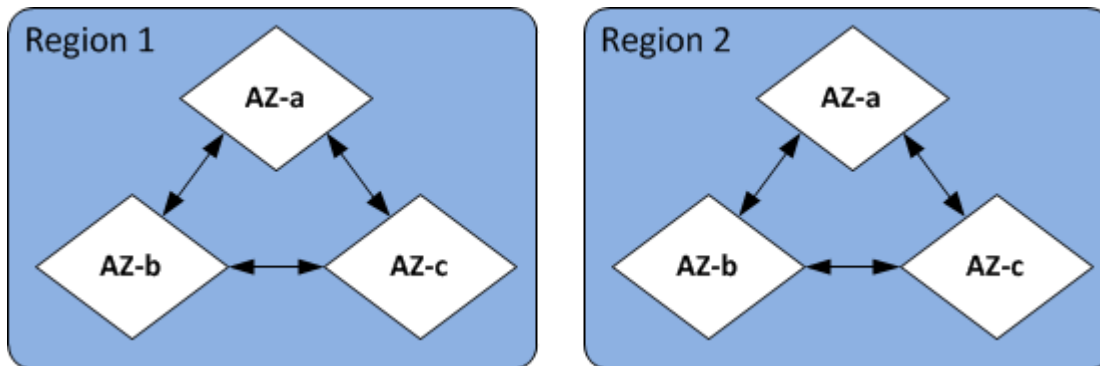
AWS Le risorse di cloud computing sono ospitate in strutture di data center ad alta disponibilità. Per fornire ulteriore scalabilità e affidabilità, queste strutture di data center sono in ubicazioni fisiche diverse. Tali località sono categorizzate in base a regioni e zone di disponibilità.

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

- [Considerazioni sulla zona di disponibilità con Memcached](#)
- [Ubicazione dei nodi](#)
- [Regioni ed endpoint supportati](#)
- [Utilizzo delle zone locali con ElastiCache](#)
- [Usare Outposts con ElastiCache](#)

Considerazioni sulla zona di disponibilità con Memcached

La distribuzione dei nodi Memcached su più zone di disponibilità all'interno di un'area consente di proteggere l'utente dall'impatto di un guasto catastrofico, ad esempio una perdita di alimentazione all'interno di una zona di disponibilità.

Cache serverless

ElastiCache la memorizzazione nella cache senza server crea una cache ad alta disponibilità che si estende su più zone di disponibilità. È possibile specificare sottoreti provenienti da zone di disponibilità diverse, allo stesso modo VPC in cui si crea il cluster serverless, oppure scegliere automaticamente le sottoreti tra quelle ElastiCache predefinite. VPC

Progettazione del proprio cluster (Memcached) ElastiCache

Un cluster Memcached può avere fino a 300 nodi. Quando si creano o si aggiungono nodi al cluster Memcached, è possibile specificare una singola zona di disponibilità per tutti i nodi, consentire

di ElastiCache scegliere una singola zona di disponibilità per tutti i nodi, specificare le zone di disponibilità per ogni nodo o consentire di scegliere una zona di disponibilità ElastiCache per ogni nodo. È possibile creare nuovi nodi in diverse zone di disponibilità quando vengono aggiunti a un cluster Memcached esistente. Una volta creato un nodo di cache, la relativa zona di disponibilità non può essere modificata.

Se desideri che un cluster in un singolo cluster di zona di disponibilità abbia i suoi nodi distribuiti su più zone di disponibilità, ElastiCache puoi creare nuovi nodi nelle varie zone di disponibilità. È quindi possibile eliminare alcuni o tutti i nodi della cache originali. Consigliamo questo approccio.

Per eseguire la migrazione dei nodi Memcached da una singola zona di disponibilità a più zone di disponibilità

1. Modificare il cluster creando nuovi nodi della cache nelle zone di disponibilità in cui si desidera. Nella richiesta, procedere come segue:
 - Imposta AZMode (CLI:- `-az-mode`) su `cross-az`.
 - Imposta NumCacheNodes (CLI:- `-num-cache-nodes`) sul numero di nodi di cache attualmente attivi più il numero di nuovi nodi di cache che desideri creare.
 - Imposta NewAvailabilityZones (CLI:- `-new-availability-zones`) su un elenco delle zone in cui desideri creare i nuovi nodi di cache. Per consentire di ElastiCache determinare la zona di disponibilità per ogni nuovo nodo, non specificate un elenco.
 - Imposta ApplyImmediately (CLI:- `-apply-immediately`) su `true`.

Note

Se non si utilizza l'individuazione automatica, assicurarsi di aggiornare l'applicazione client con i nuovi endpoint del nodo di cache.

Prima di passare al passaggio successivo, assicurarsi che i nodi Memcached siano completamente creati e disponibili.

2. Modificare il cluster rimuovendo i nodi che non si desidera più nella zona di disponibilità originale. Nella richiesta, procedere come segue:
 - Imposta NumCacheNodes (CLI:- `-num-cache-nodes`) sul numero di nodi di cache attivi che desideri dopo l'applicazione di questa modifica.

- Imposta `CacheNodeIdsToRemove` (CLI: `-nodes-to-remove`) su un elenco dei nodi di cache che desideri rimuovere dal cluster.

Il numero di nodi di cache IDs elencati deve essere uguale al numero di nodi attualmente attivi meno il valore `inNumCacheNodes`.

- (Facoltativo) Imposta `ApplyImmediately` (CLI: `-apply-immediately`) su `true`.

Se non imposti `ApplyImmediately` (CLI: `-apply-immediately`) su `true`, le eliminazioni dei nodi avverranno nella prossima finestra di manutenzione.

Ubicazione dei nodi

Amazon ElastiCache supporta la localizzazione di tutti i nodi di un cluster in una o più zone di disponibilità (AZs). Inoltre, se scegli di localizzare i nodi in più nodi AZs (scelta consigliata), ti ElastiCache consente di scegliere la zona Z per ogni nodo o di scegliere i nodi ElastiCache al posto tuo.

Posizionando i nodi in modo diverso AZs, si elimina la possibilità che un guasto, ad esempio un'interruzione dell'alimentazione, in una zona AZ provochi il malfunzionamento dell'intero sistema. I test hanno dimostrato che non vi è alcuna differenza di latenza significativa tra l'individuazione di tutti i nodi in una zona AZ o la loro distribuzione su più nodi. AZs

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 per Memcached](#)
- [Creazione di un cluster per Valkey o Redis OSS](#)
- [Modifica di un cluster ElastiCache](#)
- [Aggiungere nodi a un ElastiCache 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 luoghi diversi per ottenere la tolleranza AZs agli errori. Per ulteriori informazioni sulle regioni e sulle zone di disponibilità, consulta [Scelta delle regioni e delle zone di disponibilità per ElastiCache](#) all'inizio di questo argomento.

Regioni in cui ElastiCache è supportato

Nome regione/R egione	Endpoint	Protocollo	
Stati Uniti orientali (Ohio) us-east-2	elasticache.us- east-2.amazo naws.com	HTTPS	
Stati Uniti orientali (Virginia settentri onale) us-east-1	elasticache.us- east-1.amazo naws.com	HTTPS	
Regione Stati Uniti occidentali (California settentrionale) us-west-1	elasticache.us- west-1.amazo naws.com	HTTPS	
Stati Uniti occidentali (Oregon) us-west-2	elasticache.us- west-2.amazo naws.com	HTTPS	
Regione Canada (Centrale) ca-central-1	elasticache.ca- central-1.am azonaws.com	HTTPS	
Regione Canada (Occidentale) ca-west-1	elasticache.ca- west-1.amazo naws.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	
Regione 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 (UAE) me-central-1	elasticache.me-central-1.amazonaws.com	HTTPS	
Regione 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 delle zone locali con ElastiCache

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 a una Local Zones creando una nuova sottorete e assegnandola alla Local Zone. Quando si crea una sottorete in una zona locale, la vostra VPC viene estesa a quella zona locale. La sottorete nella zona locale funziona allo stesso modo delle altre sottoreti della zona locale. VPC

Utilizzando Local Zones, puoi collocare risorse come un ElastiCache cluster in più posizioni vicine ai tuoi utenti.

Quando si crea un ElastiCache cluster, 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 AWS regionale 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 alle ElastiCache 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. Abilita la zona locale nella EC2 console Amazon.

Per ulteriori informazioni, consulta [Enabling Local Zones](#) nella Amazon EC2 User Guide.

2. Creare una sottorete nella zona locale.

Per ulteriori informazioni, consulta [Creating a subnet in your VPC](#) nella Amazon VPC User Guide.

3. Crea un gruppo di ElastiCache sottoreti nella zona locale.

Quando crei un gruppo di ElastiCache sottoreti, scegli il gruppo di zone di disponibilità per la zona locale.

Per ulteriori informazioni, consulta [Creazione di un gruppo di sottoreti](#).

4. Crea un cluster ElastiCache (Memcached) che utilizza la ElastiCache sottorete nella zona locale.

Per ulteriori informazioni, consulta [Creazione di un cluster Memcached \(console\)](#).

5. Crea un cluster ElastiCache (RedisOSS) che utilizza la ElastiCache sottorete nella zona locale.
Per ulteriori informazioni, consulta uno dei seguenti argomenti:

- [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)
- [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)

Usare Outposts con ElastiCache

Puoi usare AWS Outposts con ElastiCache Outposts è un servizio completamente gestito che estende l'AWS infrastruttura APIs, i servizi e gli strumenti alle sedi dei clienti. Fornendo l'accesso locale all'infrastruttura AWS gestita, AWS Outposts consente ai clienti di creare ed eseguire applicazioni in locale utilizzando le stesse interfacce di programmazione AWS delle regioni, utilizzando al contempo risorse di calcolo e archiviazione locali per esigenze di latenza inferiore ed elaborazione locale dei dati. Un Outpost è un pool di capacità di AWS elaborazione e archiviazione distribuito presso la sede di un cliente. AWS gestisce, monitora e gestisce questa capacità come parte di una regione. AWS Puoi creare sottoreti su Outpost e specificarle quando crei AWS risorse come i cluster ElastiCache

Note

In questa versione si applicano le seguenti limitazioni:

- ElastiCache for Outposts supporta solo le famiglie di nodi M5 e R5.
- Multi-AZ (la replica cross-outpost non è supportata).
- La migrazione live non è supportata.
- Gli snapshot locali non sono supportati.
- I registri del motore e gli slow log non possono essere abilitati.
- ElastiCache on Outposts non supporta CoIP.

- ElastiCache for Outposts non è supportato nelle seguenti regioni: cn-north-1, cn-northwest-1 e ap-northeast-3.

Usare Outposts con la console ElastiCache

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Cache Valkey, cache Redis o OSS cache Memcached.
3. Se hai scelto le cache Valkey, seleziona Crea cache Valkey. Se hai scelto le cache Redis, seleziona OSS Crea cache Redis. OSS Se hai scelto le cache Memcached, seleziona Crea cache Memcached.
4. In Impostazioni cluster, seleziona Progetta la tua cache e Cluster cache. Lascia la modalità Cluster impostata su Disabilitata. Quindi crea un nome e una descrizione opzionale per la cache.
5. Per la posizione, scegli On premises.
6. Nella sezione Locale vedrai il campo Outpost ID. Inserisci l'ID per il luogo in cui verrà eseguito il cluster.

Tutte le altre impostazioni in Impostazioni cluster possono rimanere come predefinite.

7. In Connettività, seleziona Crea un nuovo gruppo di sottoreti e inserisci l'VPCID. Lascia il resto come predefinito e seleziona Avanti.

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 Valkey, Impostazioni Redis o OSS Impostazioni Memcached, a seconda del motore preferito:
 - a. Nome: inserisci un nome per il cluster
 - b. Descrizione: inserire una descrizione per il cluster.
 - c. Compatibilità della versione del motore: la versione del motore è basata sulla regione Outpost AWS

- d. Porta: per Valkey o RedisOSS, accetta la porta predefinita 6379. Per Memcached, accetta la porta predefinita 11211. Se preferisci usare una porta diversa, digita il numero di 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. Se usi Valkey o RedisOSS, Porting Assistant for. NETfor 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 Connettività:

- 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.
 - VPCID: l'VPCID deve corrispondere all'OutpostVPC. Se ne selezioni una VPC che non ha alcuna sottorete IDs negli Outposts, l'elenco risulterà 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 ci sono sottoreti IDs disponibili, devi crearle. Per ulteriori dettagli, vedi [Crea una sottorete](#).
- b. Seleziona Crea.

Visualizzazione dei dettagli del cluster Outpost

Nella pagina dell'elenco, seleziona un cluster che appartiene a un AWS Outpost e tieni presente quanto segue quando visualizzi i dettagli del cluster:

- Zona di disponibilità: rappresenterà l'avamposto, utilizzando un ARN (Amazon Resource Name) e il AWS Resource Number.
- Nome dell'avamposto: il nome dell'avamposto. AWS

Usare Outposts con AWS CLI

È possibile utilizzare il comando AWS Command Line Interface (AWS CLI) per controllare più AWS servizi dalla riga di comando e automatizzarli tramite script. È possibile utilizzare il AWS CLI per operazioni ad hoc (una tantum).

Scaricamento e configurazione di AWS CLI

AWS CLI Funziona su Windows, macOS o Linux. Utilizzare la procedura seguente per scaricarla e configurarla.

Per scaricare, installare e configurare CLI

1. Scaricate il AWS CLI file dalla pagina Web dell'[interfaccia a riga di AWS comando](#).
2. Segui le istruzioni per [l'installazione AWS CLI e la configurazione riportate AWS CLI nella Guida per l'AWS Command Line Interface utente](#).

Usare AWS CLI con Outposts

Usa la seguente CLI operazione per creare un cluster di cache che utilizza Outposts:

- [create-cache-cluster](#)— Utilizzando questa operazione, il `outpost-mode` parametro accetta un valore che specifica se i nodi nel cluster di cache vengono creati in un singolo Outpost o tra 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 della tua ElastiCache implementazione.

Argomenti

- [Snapshot e ripristino](#)
- [Versioni del motore e aggiornamento in ElastiCache](#)
- [ElastiCache migliori pratiche e strategie di caching](#)
- [Gestione del cluster progettato autonomamente in ElastiCache](#)
- [Ridimensionamento ElastiCache](#)
- [Guida introduttiva JSON a Valkey e Redis OSS](#)
- [Taggare le tue risorse ElastiCache](#)
- [Utilizzo dell'obiettivo Amazon ElastiCache Well-Architected](#)
- [Procedure di risoluzione dei problemi e procedure consigliate comuni con ElastiCache](#)

Snapshot e ripristino

ElastiCache Le cache Amazon che eseguono Valkey, Redis OSS o Serverless Memcached possono eseguire 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 tuoi dati creando una nuova cache Memcached Valkey, Redis OSS o Serverless e popolandola con i dati di un backup. Con ElastiCache, puoi gestire i backup utilizzando il, il () e AWS Management Console il. AWS Command Line Interface AWS CLI ElastiCache API

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](#)
- [Tutorial: seminare 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:

- Il backup e il ripristino sono supportati solo per le cache in esecuzione su Valkey, Redis OSS o Serverless Memcached.
- Per i cluster Valkey o Redis OSS (modalità cluster disabilitata), il backup e il ripristino non sono supportati sui nodi. `cache.t1.micro` Tutti gli altri tipi di nodi di cache sono supportati.
- Per i cluster Valkey o Redis OSS (modalità cluster abilitata), il backup e il ripristino sono supportati per tutti i tipi di nodi.
- Durante un periodo contiguo di 24 ore, è possibile creare non più di 24 backup manuali per cache serverless. Per i cluster OSS progettati autonomamente da Valkey e Redis, è possibile creare non più di 20 backup manuali per nodo del cluster.
- Valkey o Redis OSS (modalità cluster abilitata) supportano solo l'esecuzione di backup a livello di cluster (a livello o, a livello di gruppo di replica). API CLI Valkey o Redis OSS (modalità cluster abilitata) non supportano l'esecuzione di backup a livello di shard (per o, a livello di gruppo di nodi).
API CLI
- Durante il processo di backup, non è possibile eseguire altre CLI operazioni sulla cache API serverless. È possibile eseguire API le nostre CLI operazioni su un cluster progettato autonomamente durante il backup.

- Se utilizzi OSS cache Valkey o Redis con tiering dei dati, non puoi esportare un backup su 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. I backup per cluster progettati autonomamente non sono disponibili con ElastiCache (Memcached) ma sono disponibili con (Redis). ElastiCache OSS

Di seguito sono elencate le linee guida per migliorare le prestazioni di backup dei cluster progettati autonomamente.

- Imposta il `reserved-memory-percent` parametro: per ridurre il paging eccessivo, si consiglia di impostare il parametro `reserved-memory-percent`. Questo parametro impedisce a Valkey e Redis OSS di consumare tutta la memoria disponibile del nodo e può aiutare a ridurre la quantità di paging. È anche possibile che si verifichino miglioramenti delle prestazioni utilizzando semplicemente un nodo più grande. Per ulteriori informazioni sulla memoria riservata e sui parametri, vedere [reserved-memory-percent](#) [Gestione della memoria riservata per Valkey e Redis OSS](#)
- Creare backup da una replica di lettura: se si esegue Valkey o Redis OSS in un gruppo di nodi con più di un nodo, è possibile eseguire un backup dal nodo primario o da una delle repliche di lettura. A causa delle risorse di sistema necessarie durante il processo BGSAVE, si consiglia di creare backup da una delle repliche di lettura. Durante la creazione del backup dalla replica, il nodo primario rimane inalterato dai requisiti di risorse. 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 si elimina un gruppo di replica e si richiede un backup finale, esegue ElastiCache sempre il backup dal nodo principale. Ciò garantisce l'acquisizione dei OSS dati Valkey o Redis più recenti, prima che il gruppo di replica venga eliminato.

Pianificazione di backup automatici

Puoi abilitare i backup automatici per qualsiasi cache OSS serverless Valkey o Redis o cluster progettato autonomamente. Quando i backup automatici sono abilitati, ElastiCache crea un backup della cache su base giornaliera. 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](#).

È possibile abilitare i backup automatici per qualsiasi cache Memcached Serverless. Quando i backup automatici sono abilitati, ElastiCache crea un backup della cache su base giornaliera. 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: un'ora del giorno in cui ElastiCache inizia la creazione di un backup. Puoi impostare la finestra di backup in qualsiasi momento e quando è più comodo. Se non si specifica una finestra di backup, ne ElastiCache assegna una automaticamente.
- 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.

Quando pianifichi i backup automatici, ElastiCache inizierà a creare il backup. Puoi impostare la finestra di backup in qualsiasi momento e quando è più comodo. Se non si specifica una finestra di backup, ne ElastiCache assegna una automaticamente.

È possibile abilitare o disabilitare i backup automatici quando si crea una nuova cache o si aggiorna una cache esistente, utilizzando la ElastiCache console, il AWS CLI, o il ElastiCache API Per Valkey

e RedisOSS, questa operazione viene eseguita selezionando la casella **Abilita backup automatici** nella sezione **Impostazioni avanzate Valkey** o **Impostazioni Redis avanzate**. OSS Per Memcached, questa operazione viene eseguita selezionando la casella **Abilita backup automatici** nella sezione **Impostazioni avanzate Memcached**.

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.

È possibile creare un backup manuale di una cache utilizzando il AWS Management Console, il AWS CLI, o il ElastiCache API.

Creazione di un backup manuale (Console)

Per creare un backup di una cache (console)

1. Accedi a AWS Management Console e apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Dal pannello di navigazione, scegli le cache Valkey, le cache Redis o le OSS cache Memcached, a seconda delle tue preferenze.
3. Scegli la casella a sinistra del nome della cache di cui desideri 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 AWS CLI

Per creare un backup manuale di una cache utilizzando il AWS CLI, utilizzare l'`create-serverless-snapshot` AWS CLI operazione 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 AWS CLI

Per creare un backup manuale di un cluster progettato autonomamente utilizzando il AWS CLI, utilizzare l'`create-snapshot` AWS CLI operazione con i seguenti parametri:

- `--cache-cluster-id`
 - Se il cluster di cui stai eseguendo il backup non ha 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*.

Utilizzate questo parametro solo per il backup di un cluster Valkey o Redis OSS (modalità cluster disabilitata).

- `--replication-group-id`— Nome del cluster Valkey o Redis OSS (modalità cluster abilitata) (CLI/API: un gruppo di replica) da utilizzare come origine per il backup. Utilizzate questo parametro per il backup di un cluster Valkey o Redis OSS (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 Valkey o Redis OSS (modalità cluster disabilitata) senza nodi di replica

La seguente AWS CLI operazione crea il backup `bkup-20150515` dal cluster Valkey o Redis OSS (modalità cluster disabilitata) `myNonClusteredRedis` che non ha 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 Valkey o Redis OSS (modalità cluster disabilitata) con nodi di replica

La seguente AWS CLI operazione crea il backup `bkup-20150515` dal cluster Valkey o Redis OSS (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
```

Esempio di output: backup di un cluster Valkey o Redis OSS (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 Valkey o Redis (modalità cluster abilitata) OSS

La seguente AWS CLI operazione crea il backup `bkup-20150515` dal cluster Valkey o Redis OSS (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

```

Esempio di output: backup di un cluster Valkey o Redis OSS (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

È possibile creare un backup finale utilizzando la ElastiCache console, il AWS CLI, o il ElastiCache API.

Creazione di un backup finale (console)

È possibile creare un backup finale eliminando una cache OSS serverless Valkey o Redis, un cluster OSS progettato autonomamente da Valkey o Redis o una cache serverless Memcached utilizzando la console. ElastiCache

Per creare un backup finale quando si elimina una cache, nella finestra di dialogo di eliminazione scegli Sì in Crea backup e assegna un nome al backup.

Argomenti correlati

- [Utilizzando il AWS Management Console](#)
- [Eliminazione di un gruppo di replica \(console\)](#)

Creazione di un backup finale (AWS CLI)

È possibile creare un backup finale quando si elimina una cache utilizzando il. AWS CLI

Argomenti

- [Quando si elimina una cache Valkey, una cache Redis o una cache serverless OSS Memcached](#)
- [Quando si elimina un OSS cluster Valkey o Redis senza repliche di lettura](#)
- [Quando si elimina un OSS cluster Valkey o Redis con repliche di lettura](#)

Quando si elimina una cache Valkey, una cache Redis o una cache serverless OSS Memcached

Per creare un backup finale, utilizzare l'operazione con i `delete-serverless-cache` AWS CLI 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, vedere [delete-serverless-cache](#) nel AWS CLI Command Reference.

Quando si elimina un OSS cluster Valkey o Redis senza repliche di lettura

Per creare un backup finale per un cluster progettato autonomamente senza repliche di lettura, utilizza l'operazione con i `delete-cache-cluster` AWS CLI 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 CLI Command Reference.

Quando si elimina un OSS cluster Valkey o Redis con repliche di lettura

Per creare un backup finale quando si elimina un gruppo di replica, utilizzate l'`delete-replication-group` AWS CLI operazione 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, vedere [delete-replication-group](#) nel AWS CLI Command Reference.

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 utilizzando il AWS Management Console

1. Accedi a AWS Management Console e apri la ElastiCache console 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, facoltativamente, dettagli su un backup specifico, utilizza l'operazione. `describe-serverless-cache-snapshots` 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, vedere [describe-serverless-cache-snapshots](#) nel Command Reference.

AWS CLI

Descrizione dei backup di cluster progettati autonomamente (AWS CLI)

Per visualizzare un elenco di backup del cluster progettati autonomamente e, facoltativamente, dettagli su un backup specifico, utilizza l'operazione. `describe-snapshots` 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-snapshots](#) nel Command Reference. AWS CLI

Copia di backup

Puoi creare una copia di qualsiasi backup, a prescindere che sia stato eseguito automaticamente o manualmente. Puoi anche esportare il backup in modo da potervi accedere dall'esterno 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. Accedi a AWS Management Console e apri la ElastiCache console 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 \  
  --target-serverless-cache-id my-cache-2023-11-27-03-15
```

```
--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 backup ElastiCache (RedisOSS) in un bucket Amazon Simple Storage Service (Amazon S3), che consente di accedervi dall'esterno. ElastiCache Puoi esportare un backup utilizzando la ElastiCache console, il AWS CLI o. ElastiCache API

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 seminare la nuova cache invece di attendere che il nuovo cluster venga popolato tramite l'uso. Per informazioni sull'inizializzazione di un nuovo cluster, consulta [Tutorial: seminare un nuovo cluster progettato autonomamente con un backup creato esternamente](#). Un altro motivo per cui potresti voler esportare i dati della cache è 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 [Suddivisione dei dati su più livelli in ElastiCache](#).
- L'esportazione di un backup è disponibile per i cluster OSS progettati autonomamente da Valkey e Redis, Serverless Valkey e Redis e Serverless Memcached. OSS L'esportazione di un backup non è disponibile per i cluster Memcached progettati autonomamente.

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.

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/>
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 -compiacente. DNS Altrimenti, non ElastiCache potrai accedere al tuo file di backup. Le regole di DNS conformità 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 Create (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.

Concedi ElastiCache l'accesso al tuo 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 vivamente di

impostare IAM politiche 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' ElastiCache accesso a un bucket S3

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo. <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 [Creazione di un bucket Amazon S3](#).
3. Scegli la scheda Autorizzazioni e in Autorizzazioni, scegli Elenco di controllo degli accessi (ACL), quindi scegli Modifica.
4. Aggiungi l'ID canonico
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
all'assegnatario con le opzioni seguenti:
 - Oggetti: Elenco, Scrittura
 - BucketACL: leggi, scrivi

Note

- Per la PDT GovCloud Regione, l'ID canonico è.
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Per la OSU GovCloud Regione, l'ID canonico è.
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c

5. Seleziona Salva.

Esportazione e backup ElastiCache

Ora hai creato il tuo bucket S3 e concesso ElastiCache le autorizzazioni per accedervi.

Successivamente, puoi utilizzare la ElastiCache console AWS CLI, o il ElastiCache API per esportare la tua istantanea al suo interno. Gli esempi seguenti presuppongono che l'IAM identità del chiamante disponga delle seguenti autorizzazioni aggiuntive specifiche per S3. IAM

```
{
  "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 a AWS Management Console e apri la ElastiCache console all'indirizzo. <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 1.000 caratteri e può essere codificato 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 [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 [Concedi ElastiCache l'accesso al tuo 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 [Concedi ElastiCache l'accesso al tuo bucket Amazon S3](#) per aggiungere le autorizzazioni specificate e riprovare a esportare il backup.

- ElastiCache non sono state concesse le READ autorizzazioni %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di lettura sul bucket.

- ElastiCache non ha ricevuto le WRITE autorizzazioni %s sul bucket S3.

Soluzione: aggiungere autorizzazioni di scrittura sul bucket.

- ElastiCache non sono state concesse READ _ ACP autorizzazioni %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'`export-serverless-cache-snapshot`CLIoperazione 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'copy-snapshotCLI operazione 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 1.000 caratteri e può essere codificato 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 [Concedi ElastiCache l'accesso al tuo 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 da Valkey in una nuova cache Valkey o in un cluster progettato autonomamente e ripristinare un OSS backup Redis esistente in una nuova cache Redis o in un cluster progettato autonomamente. OSS È inoltre possibile ripristinare un backup della cache serverless Memcached esistente in una nuova cache serverless Memcached.

Ripristino di un backup in una cache serverless (console)

Note

ElastiCache Serverless supporta RDB file compatibili con Valkey 7.2 e versioni successive e versioni Redis tra la 5.0 e l'ultima versione disponibile. OSS

Per ripristinare un backup in una cache serverless (console)

1. Accedi a AWS Management Console e apri la console all'indirizzo. 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 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 a AWS Management Console e apri la ElastiCache console 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 RDB file compatibili con Valkey 7.2 e versioni successive e OSS versioni Redis tra la 5.0 e l'ultima versione disponibile.

Per ripristinare un backup in una nuova cache serverless (AWS CLI)

L' AWS CLI esempio seguente 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)

È possibile ripristinare un backup Valkey o Redis OSS (modalità cluster disabilitata) in due modi.

- ```
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)

È possibile ripristinare un backup della cache OSS senza server Valkey o Redis e ripristinare anche un cluster progettato autonomamente da Valkey o Redis. OSS

È possibile ripristinare un backup della cache serverless Valkey o Redis OSS in due modi.

- È possibile eseguire il ripristino in un cluster Valkey o Redis a nodo singolo OSS (modalità cluster disabilitata) utilizzando l'operazione. AWS CLI create-cache-cluster
- È possibile eseguire il ripristino in un OSS cluster Valkey o Redis con repliche di lettura (un gruppo di replica). A tale scopo, è possibile utilizzare Valkey o Redis OSS (modalità cluster disabilitata) o Valkey o Redis OSS (modalità cluster abilitata) con l'operazione. AWS CLI create-replication-group In questo caso, si esegue il ripristino con un file Valkey o Redis .rdb. OSS Per ulteriori informazioni sull'inizializzazione di un nuovo cluster progettato autonomamente, consulta [Tutorial: seminare un nuovo cluster progettato autonomamente con un backup creato esternamente](#).

È possibile ripristinare un backup Valkey o Redis OSS (modalità cluster disabilitata) in due modi.

- È possibile eseguire il ripristino su un cluster Valkey o Redis a nodo singolo OSS (modalità cluster disabilitata) utilizzando l'operazione. AWS CLI create-cache-cluster
- È possibile eseguire il ripristino in un OSS cluster Valkey o Redis con repliche di lettura (un gruppo di replica). A tale scopo, è possibile utilizzare Valkey o Redis OSS (modalità cluster disabilitata) o Valkey o Redis OSS (modalità cluster abilitata) con l'operazione. AWS CLI create-replication-group In questo caso, si esegue il ripristino con un file Valkey o Redis .rdb. OSS Per ulteriori informazioni

sull'inizializzazione di un nuovo cluster progettato autonomamente, consulta [Tutorial: seminare 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'API operazione di eliminazione che consente di eliminare un backup in qualsiasi momento, indipendentemente dal fatto 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.

È possibile eliminare un backup utilizzando la ElastiCache console AWS CLI, il o ElastiCache API.

### Eliminazione di un backup (console)

La procedura seguente elimina un backup utilizzando la ElastiCache console.

Per eliminare un backup

1. Accedi a AWS Management Console e apri la ElastiCache console 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. Scegli 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)

Utilizzate l' AWS CLI operazione `delete-snapshot` con il seguente parametro per eliminare un backup senza server.

- `--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, vedere [delete-serverless-cache-snapshot](#) nel Command Reference.AWS CLI

## Eliminazione di un backup del cluster progettato autonomamente (AWS CLI)

Utilizzate l' AWS CLI operazione `delete-snapshot` 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 di allocazione dei costi sono un mezzo per tenere traccia dei costi su più AWS servizi raggruppando le spese sulle fatture in base ai valori dei tag. Per ulteriori informazioni sui tag di allocazione dei costi, consulta la sezione relativa all'[uso dei tag di allocazione dei costi](#).

Utilizzando la ElastiCache console, oppure è ElastiCache API possibile aggiungere AWS CLI, elencare, modificare, rimuovere o copiare i tag di allocazione dei costi sui backup. Per ulteriori informazioni, consulta [Monitoraggio dei costi con i tag di allocazione dei costi](#).

## Tutorial: seminare un nuovo cluster progettato autonomamente con un backup creato esternamente

Quando crei un nuovo cluster OSS progettato autonomamente da Valkey o Redis, puoi eseguire il seed con i dati di un file di backup Valkey o Redis .rdb. OSS Il seeding del cluster è utile se attualmente gestisci un'OSSistanza Valkey o Redis all'esterno ElastiCache e desideri popolare il tuo nuovo cluster ElastiCache (Redis) progettato autonomamente con i dati Valkey o OSS Redis esistenti. OSS

Per eseguire il seeding di un nuovo cluster OSS progettato autonomamente da Valkey o Redis da un backup Valkey o Redis OSS creato all'interno di Amazon, vedi. ElastiCache [Ripristino da un backup in una nuova cache](#)

Quando usi un file Valkey o Redis OSS .rdb per seminare un nuovo cluster progettato autonomamente, puoi fare quanto segue:

- Esegui l'aggiornamento da un cluster non partizionato a un cluster autoprogettato Valkey o Redis OSS (modalità cluster abilitata) che esegue Redis versione 3.2.4. OSS
- Specificate un numero di shard (chiamati gruppi di nodi in and) nel nuovo cluster progettato autonomamente. API CLI 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 la scalabilità a un tipo di nodo più piccolo, assicurati che il nuovo tipo di nodo abbia memoria sufficiente per i dati e il sovraccarico di Valkey o Redis. OSS Per ulteriori informazioni, consulta [Assicurarsi di disporre di memoria sufficiente per creare un'istantanea Valkey o Redis OSS](#).
- Distribuisci le chiavi negli slot del nuovo cluster Valkey o Redis OSS (modalità cluster abilitata) in modo diverso rispetto al cluster utilizzato per creare il file di backup.

### Note

Non è possibile eseguire il seeding di un cluster Valkey o Redis OSS (modalità cluster disabilitata) da un file.rdb creato da un cluster Valkey o OSS Redis (modalità cluster abilitata).

**⚠ Important**

- È necessario assicurarsi che i dati di OSS backup di Valkey o Redis non superino le risorse del nodo. Ad esempio, non puoi caricare un file.rdb con 5 GB di OSS dati Valkey o Redis su un nodo cache.m3.medium con 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 dei tipi e delle specifiche dei nodi, consulta le [caratteristiche Parametri specifici del tipo di nodo Redis OSS e i dettagli dei ElastiCache prodotti Amazon](#).

- Puoi crittografare un file Valkey o Redis OSS .rdb solo con la crittografia lato server di Amazon S3 (-S3). SSE Per ulteriori informazioni, consulta [Protezione dei dati con la crittografia lato server](#).

Di seguito, puoi trovare argomenti che ti guidano nella migrazione del cluster dall'esterno per Valkey o Redis a (Redis). ElastiCache OSS ElastiCache OSS

**Migrazione a (Redis) ElastiCache OSS**

- [Passaggio 1: creare un backup Valkey o Redis OSS](#)
- [Fase 2: creazione di un bucket Amazon S3 e una cartella](#)
- [Fase 3: caricamento del backup in Amazon S3](#)
- [Passaggio 4: concedere l'accesso in ElastiCache lettura al file.rdb](#)

**Migrazione da servizi esterni a ElastiCache (Redis). OSS**

- [Passaggio 1: creare un backup Valkey o Redis OSS](#)
- [Fase 2: creazione di un bucket Amazon S3 e una cartella](#)
- [Fase 3: caricamento del backup in Amazon S3](#)
- [Passaggio 4: concedere l'accesso in ElastiCache lettura al file.rdb](#)

## Passaggio 1: creare un backup Valkey o Redis OSS

Per creare il OSS backup Valkey o Redis per eseguire il seeding dell'istanza (Redis) ElastiCache OSS

1. Connect alla tua istanza Valkey o Redis OSS esistente.
2. Esegui una delle due BGSAVE SAVE operazioni 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, vedere il [BGSAVE](#) sito Web di Valkey.

SAVE è sincrona e blocca altri processi finché non è terminata. Per ulteriori informazioni, vedere il sito [SAVE](#) Web di Valkey.

Per ulteriori informazioni sulla creazione di un backup, vedere [Persistence](#) at the Valkey Web.

## 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 a 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 -compiacente. DNS Altrimenti, non ElastiCache potrai accedere al tuo file di backup. Le regole di DNS conformità 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).

È necessario creare il bucket Amazon S3 nella stessa AWS regione del nuovo cluster ElastiCache (RedisOSS). Questo approccio garantisce la massima velocità di trasferimento dei dati durante la ElastiCache lettura del file.rdb da Amazon S3.

#### Note

Per tenere i dati al sicuro, crea autorizzazioni al bucket Amazon S3 il più possibile restrittive. Allo stesso tempo, le autorizzazioni devono comunque consentire l'utilizzo del bucket e del suo contenuto per il seeding del nuovo cluster Valkey o Redis. OSS

Per aggiungere una cartella a un bucket Amazon S3

1. Accedi a AWS Management Console e apri la console 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 [Passaggio 1: creare un backup Valkey o Redis OSS](#). 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 a 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.

#### Passaggio 4: concedere l'accesso in ElastiCache lettura al file.rdb

Ora concedi l'accesso in ElastiCache lettura al tuo file di backup.rdb. Concedi ElastiCache l'accesso al tuo file di backup in un modo diverso a seconda che il bucket si trovi in una AWS regione predefinita o in una regione opzionale. AWS

AWS Le regioni introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. Puoi iniziare a lavorare in queste AWS regioni immediatamente. Le regioni introdotte dopo il 20 marzo 2019, 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.

Scegli il tuo approccio in base alla tua AWS regione:

- Per una regione predefinita, utilizzare la procedura in [Concedi l'accesso in ElastiCache lettura al file.rdb in una regione predefinita](#).

- Per una regione scelta, utilizzare la procedura in [Concedi l'accesso in ElastiCache lettura al file.rdb in una regione opzionale](#).

## Concedi l'accesso in ElastiCache lettura al file.rdb in una regione predefinita

AWS Le regioni introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. Puoi iniziare a lavorare in queste AWS regioni immediatamente. Le regioni introdotte dopo il 20 marzo 2019, 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.

Per concedere l'accesso in ElastiCache lettura al file di backup in una AWS regione è abilitato per impostazione predefinita

1. Accedi a 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 dei canonical IDs del seguente elenco non è elencato come utente, procedi come segue:
  - a. In Accesso per altri AWS account, scegli Aggiungi beneficiario.
  - b. Nella casella, aggiungi l'ID canonico della AWS regione come mostrato di seguito:
    - AWS GovCloud Regione (Stati Uniti occidentali):

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

### Important

Il backup deve trovarsi in un bucket S3 AWS GovCloud (US) per poterlo scaricare su un cluster Valkey o Redis. OSS AWS GovCloud (US)

- AWS Regioni abilitate per impostazione predefinita:

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

- c. Impostare le autorizzazioni sul bucket scegliendo Yes (Sì) per:
    - Oggetto elenco/scrittura
    - Autorizzazioni di lettura/scrittura degli oggetti ACL
  - d. Seleziona Salva.
7. Scegliere Overview (Panoramica), quindi Download (Scarica).

Concedi l'accesso in ElastiCache lettura al file.rdb in una regione opzionale

AWS Le regioni introdotte prima del 20 marzo 2019 sono abilitate per impostazione predefinita. Puoi iniziare a lavorare in queste AWS regioni immediatamente. Le regioni introdotte dopo il 20 marzo 2019, 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 concedi l'accesso in ElastiCache lettura al tuo file di backup.rdb.

Per concedere l'accesso in ElastiCache lettura al file di backup

1. Accedi a 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 Autorizzazioni.
6. Sotto Autorizzazioni, scegli Policy bucket e seleziona Modifica.
7. Aggiorna la policy per concedere le autorizzazioni ElastiCache 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. Scegli Save changes (Salva modifiche).

Semina il ElastiCache cluster con i dati del file.rdb

Ora sei pronto per creare un ElastiCache cluster e seminarlo con i dati del file.rdb. Per creare il cluster, segui le istruzioni in [Creazione di un cluster per Valkey o Redis OSS](#) o [Creazione da zero di un gruppo di OSS replica Valkey o Redis](#). Assicurati di scegliere Valkey o Redis OSS come motore del cluster.

Il metodo utilizzato per indicare ElastiCache dove trovare il backup caricato su Amazon S3 dipende dal metodo utilizzato per creare il cluster:

Esegui il seme del cluster ElastiCache (RedisOSS) o del 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 Posizione S3 RDB del file di origine, digita il percorso Amazon S3 per i file. Se disponi di più file .rdb, digita il percorso per ogni file in un elenco separato da virgole. Il percorso Amazon S3 appare simile a *myBucket/myFolder/myBackupFilename*.rdb.

- Usando il AWS CLI

Se si utilizza l'create-replication-groupoperazione create-cache-cluster o, utilizzare il parametro --snapshot-arns per specificare un valore completo ARN per ogni file.rdb. Ad esempio `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. ARNDevono essere risolti nei file di backup archiviati in Amazon S3.

- Utilizzando il ElastiCache API

Se si utilizza l>CreateReplicationGroup ElastiCache APIoperazione CreateCacheCluster o, utilizzare il parametro SnapshotArns per specificare un valore completo ARN per ogni file.rdb. Ad esempio `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. ARNDevono essere risolti nei file di backup archiviati in Amazon S3.

### Important

Quando si esegue il seeding di un cluster Valkey o Redis OSS (modalità cluster abilitata), è necessario configurare ogni gruppo di nodi (shard) nel nuovo cluster o gruppo di replica. Utilizzate il parametro --node-group-configuration (API:NodeGroupConfiguration) per eseguire questa operazione. Per ulteriori informazioni, consulta gli argomenti seguenti:

- CLI: [create-replication-group](#)nel AWS CLI riferimento
- API: [CreateReplicationGroup](#)nel ElastiCache API riferimento

Durante il processo di creazione del cluster, i dati del OSS backup Valkey o Redis vengono scritti nel cluster. È possibile monitorare l'avanzamento visualizzando i messaggi degli ElastiCache eventi. Per fare ciò, consulta la ElastiCache console e scegli Cache Events. Puoi anche usare l'interfaccia a riga di AWS ElastiCache comando o ElastiCache API per ottenere messaggi di eventi. Per ulteriori informazioni, consulta [Visualizzazione ElastiCache degli eventi](#).

# Versioni del motore e aggiornamento in ElastiCache

Questa sezione descrive i motori Valkey, Redis OSS e Memcached supportati e le modalità di aggiornamento. Tieni presente che tutte le funzionalità disponibili con Redis OSS 7.2 sono disponibili in Valkey 7.2 e versioni successive per impostazione predefinita. Puoi anche eseguire l'aggiornamento da alcuni OSS motori esistenti ElastiCache con Redis a un motore Valkey.

## Argomenti

- [Gestione delle versioni per ElastiCache](#)
- [Come aggiornare la versione di un motore](#)
- [Come attivare l'aggiornamento cross-engine da Redis OSS a Valkey](#)
- [Motori e versioni supportati](#)
- [Principali differenze di comportamento e compatibilità della versione con Valkey](#)
- [Principali differenze di comportamento e compatibilità delle versioni con Redis OSS](#)
- [Risoluzione degli aggiornamenti del motore Valkey o Redis OSS bloccati](#)

## Gestione delle versioni per ElastiCache

Gestisci il modo in cui desideri aggiornare le ElastiCache cache e i cluster progettati autonomamente e aggiornati per i motori Valkey, Redis e Memcached. OSS

## Gestione delle versioni per ElastiCache Serverless Cache

Gestisci se e quando la cache ElastiCache Serverless viene aggiornata ed esegui gli aggiornamenti di versione secondo le tue condizioni e tempistiche.

ElastiCache Serverless applica automaticamente la versione più recente MINOR e quella del PATCH software alla cache, senza alcun impatto o tempi di inattività sull'applicazione. Non è richiesta nessuna azione da parte tua.

Quando è disponibile una nuova MAJOR versione, ElastiCache Serverless ti invierà una notifica nella console e un evento in. EventBridge Puoi scegliere di aggiornare la cache alla versione principale più recente modificandola utilizzando la Console o API selezionando la versione più recente del motore.

CLI

## Gestione delle versioni per cluster progettati autonomamente ElastiCache

Quando si lavora con ElastiCache cluster progettati autonomamente, è possibile controllare quando il software che alimenta il cluster di cache viene aggiornato alle nuove versioni supportate da.

ElastiCache È possibile controllare quando aggiornare la cache alle versioni e alle versioni più recenti disponibili MAJOR. MINOR PATCH 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.

È possibile controllare se e quando il software conforme al protocollo che alimenta il cluster di cache viene aggiornato alle nuove versioni supportate 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.

### Cluster Valkey e Redis OSS

#### Note

- Se un OSS cluster Valkey o Redis viene replicato in una o più regioni, la versione del motore viene aggiornata per le regioni secondarie e quindi per la regione principale.
- ElastiCache Le versioni (RedisOSS) sono identificate con una versione semantica che comprende un componente and. MAJOR MINOR Ad esempio, in Redis OSS 6.2, la versione principale è 6 e la versione secondaria 2. Quando si utilizzano cluster progettati autonomamente, ElastiCache (RedisOSS) espone anche il PATCH componente, ad esempio Redis OSS 6.2.1, e la versione della patch è 1.

MAJOR le versioni riguardano modifiche API incompatibili e MINOR le versioni riguardano nuove funzionalità aggiunte in modo retrocompatibile. PATCH le versioni riguardano correzioni di bug compatibili con le versioni precedenti e modifiche non funzionali.

Con Valkey e RedisOSS, è possibile avviare gli aggiornamenti della versione del motore nel cluster o nel gruppo di replica modificandolo e specificando una nuova versione del motore. Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).

### Memcached



Con Memcached, per eseguire l'aggiornamento a una versione più recente è necessario modificare il cluster di cache e specificare la nuova versione del motore che si desidera utilizzare. L'aggiornamento a una versione più recente di Memcached è un processo distruttivo: si perdono i dati e si inizia con una cache a freddo. Per ulteriori informazioni, consulta [Modifica di un cluster ElastiCache](#).

Quando viene eseguito l'aggiornamento da una versione precedente alla versione 1.4.33 o una successiva di Memcached, è importante tenere presente i requisiti riportati di seguito. `CreateCacheCluster` e `ModifyCacheCluster` non riescono nelle condizioni seguenti:

- Se `slab_chunk_max > max_item_size`.
- Se `max_item_size` modulo `slab_chunk_max`  $\neq 0$ .
- Se `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

Il valore `(max_cache_memory - memcached_connections_overhead)` rappresenta la memoria del nodo utilizzabile per i dati. Per ulteriori informazioni, consulta [Sovraccarico delle connessioni Memcached](#).

## 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 Serverless. ElastiCache

## Considerazioni su Valkey e Redis OSS

Quando aggiorni un cluster Valkey o Redis progettato autonomamente, considera quanto segue. OSS

- 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 una patch al cluster per conto dell'utente nell'improbabile eventualità che si verifichi una vulnerabilità critica di sicurezza nel sistema o nel software di cache.
- A partire da Valkey 7.2 e Redis OSS 6.0, ElastiCache offrirà un'unica versione per ogni versione minore, anziché offrire più versioni di patch.

- A partire dalla versione 5.0.6 OSS del motore Redis, puoi 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.
- Puoi anche aggiornare i tuoi ElastiCache cluster con versioni precedenti alla 5.0.6. Il processo in questione è lo stesso, ma può richiedere tempi di failover più lunghi durante DNS la propagazione (30 s-1 m).
- A partire da Redis OSS 7, ElastiCache supporta il passaggio tra Valkey o Redis (modalità cluster disabilitata) e Valkey o Redis OSS (modalità cluster abilitata). OSS
- Il processo di aggiornamento del motore Amazon ElastiCache (RedisOSS) è progettato per fare il massimo sforzo per conservare i dati esistenti e richiede una replica Redis OSS di successo.
- Quando si aggiorna il motore, ElastiCache interromperà le connessioni client esistenti. [Per ridurre al minimo i tempi di inattività durante gli aggiornamenti del motore, consigliamo di implementare le migliori pratiche per OSS i client Redis con tentativi di errore e backoff esponenziale, nonché le migliori pratiche per ridurre al minimo i tempi di inattività durante la manutenzione.](#)
- Non è possibile eseguire l'aggiornamento direttamente da Valkey o Redis OSS (modalità cluster disabilitata) a Valkey o Redis (modalità cluster abilitata) quando si aggiorna il motore. OSS La procedura seguente mostra come eseguire l'aggiornamento da Valkey o Redis OSS (modalità cluster disabilitata) a Valkey o Redis (modalità cluster abilitata). OSS

Per eseguire l'aggiornamento da una versione del motore Valkey o Redis OSS (modalità cluster disabilitata) a una versione del motore Valkey o OSS Redis (modalità cluster abilitata)

1. Effettua un backup del cluster o del gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Esecuzione di backup manuali](#).
2. Usa il backup per creare e seminare un cluster Valkey o Redis OSS (modalità cluster abilitata) con uno shard (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 [Tutorial: seminare un nuovo cluster progettato autonomamente con un backup creato esternamente](#).
3. Elimina il vecchio cluster o gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Eliminazione di un cluster in ElastiCache](#) o [Eliminazione di un gruppo di replica](#).
4. Ridimensiona il nuovo cluster o gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) in base al numero di shard (gruppi di nodi) di cui hai bisogno. Per ulteriori informazioni, consulta [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)

- 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 Redis singoli e OSS i cluster con Multi-AZ disabilitato, consigliamo di rendere disponibile a Redis una quantità di memoria sufficiente come descritto in. OSS [Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS](#) In condizioni simili, il nodo primario non sarà a disposizione delle richieste di servizi durante la procedura di aggiornamento.
- Per OSS i cluster Redis con Multi-AZ abilitato, consigliamo inoltre di pianificare gli aggiornamenti del motore durante i periodi di basso traffico di scrittura in entrata. Quando si esegue l'aggiornamento a Redis OSS 5.0.6 o versioni successive, il cluster primario continua a essere disponibile per le richieste di assistenza 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 ognle partizioni, 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 altrle partizioni.
- I nodi primari dei varle 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.

## Considerazioni su Memcached

Quando si aggiorna un cluster Memcached progettato autonomamente, si consideri quanto segue.

- 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 una patch al cluster per conto dell'utente nell'improbabile eventualità che si verifichi una vulnerabilità critica di sicurezza nel sistema o nel software di cache.
- Poiché il motore Memcached non prevede la persistenza, l'aggiornamento a una particolare versione è sempre un processo radicale, che cancella tutti i dati della cache nel cluster.

## Come aggiornare la versione di un motore

### Valkey e Redis OSS

Con Valkey e RedisOSS, è possibile avviare gli aggiornamenti di versione del cluster o del gruppo di replica modificandolo utilizzando la ElastiCache console, il o il e specificando una versione più recente del AWS CLI motore. ElastiCache API Per ulteriori informazioni, consulta i seguenti argomenti.

| Come modificare cluster e gruppi di replica                  |                                                  |
|--------------------------------------------------------------|--------------------------------------------------|
| Cluster                                                      | Gruppi di replica                                |
| <a href="#">Usando il ElastiCache AWS Management Console</a> | <a href="#">Usando il AWS Management Console</a> |
| <a href="#">Usare il con AWS CLI ElastiCache</a>             | <a href="#">Utilizzo del AWS CLI</a>             |
| <a href="#">Utilizzo del ElastiCache API</a>                 | <a href="#">Usando il ElastiCache API</a>        |

### Memcached

Con Memcached, per avviare gli aggiornamenti di versione del cluster, è necessario modificarlo e specificare una versione più recente del motore. È possibile farlo utilizzando la ElastiCache console, il AWS CLI o: ElastiCache API

- Per usare il AWS Management Console, vedi —[Usando il ElastiCache AWS Management Console](#).
- Per usare il AWS CLI, vedi [Usare il con AWS CLI ElastiCache](#).
- Per usare il ElastiCache API, vedi [Utilizzo del ElastiCache API](#).

## Come attivare l'aggiornamento cross-engine da Redis OSS a Valkey

È possibile aggiornare un gruppo di OSS replica Redis esistente (v4 o successivo) al motore Valkey utilizzando Console, oppure. API CLI

**Note**

Se desideri aggiornare un cluster Redis a nodo singolo esistente OSS (modalità cluster disabilitata) al motore Valkey, devi prima aggiungerlo a un gruppo di replica seguendo questi passaggi preliminari: [Creazione di un gruppo di replica utilizzando un cluster esistente](#)

Se disponi di un gruppo di OSS replica Redis esistente che utilizza il gruppo di parametri di cache predefinito, puoi eseguire l'aggiornamento a Valkey specificando il nuovo motore e la versione del motore con. `modify-replication-group` API

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --engine valkey \
 --engine-version 7.2
```

Per Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --engine valkey ^
 --engine-version 7.2
```

Se hai un gruppo di parametri di cache personalizzato applicato al gruppo di replica redis esistente che desideri aggiornare, dovrai anche passare un gruppo di parametri di cache Valkey personalizzato nella richiesta. Il gruppo di parametri personalizzati Valkey di input deve avere gli stessi valori di parametri statici Redis del gruppo di parametri personalizzati Redis esistente.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --engine valkey \
 --engine-version 7.2 \
 --cache-parameter-group-name myParamGroup
```

Per Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --engine valkey ^
 --engine-version 7.2 ^
 --cache-parameter-group-name myParamGroup
```

## Aggiornamento multimotore da Redis OSS a Valkey per Serverless ElastiCache

È possibile aggiornare una cache OSS serverless Redis esistente al motore Valkey utilizzando la Console API oppure CLI, specificando il nuovo motore e la versione principale del motore con. `modify-serverless-cache` API

Per Linux, macOS o Unix:

```
aws elasticache modify-serverless-cache \
 --serverless-cache-name myCluster \
 --engine valkey \
 --major-engine-version 7
```

Per Windows:

```
aws elasticache modify-serverless-cache ^
 --serverless-cache-name myCluster ^
 --engine valkey ^
 --major-engine-version 7
```

## Motori e versioni supportati

ElastiCache Le cache serverless supportano Valkey 7.2 e versioni successive, Redis OSS versione 7.0 e Memcached 1.6 e successive.

ElastiCache le cache autoprogettate supportano Valkey 7.2 e versioni successive, tutte le versioni OSS Redis 4.0.10 e successive e le versioni Memcached 1.4.5 e successive.

ElastiCache I cluster progettati autonomamente supportano le seguenti versioni di Valkey:

- [Versioni Valkey supportate](#)
- [Versioni Redis supportate OSS](#)
- [Pianificazione di fine vita delle versioni Redis OSS](#)
- [Versioni supportate ElastiCache \(Memcached\)](#)

## Versioni Valkey supportate

Versioni Valkey supportate di seguito. Nota che Valkey supporta la maggior parte delle funzionalità disponibili in Redis OSS 7.2 per impostazione predefinita.

### ElastiCache (Valkey) versione 7.2.6

Il 10 ottobre 2024, ElastiCache con Valkey 7.2.6 è stato rilasciato. Ecco alcune delle nuove funzionalità introdotte in Valkey 7.2 (rispetto a Redis 7.1): OSS

- Nuova opzione WITHSCORE e comandi ZRANK ZREVRANK
- CLIENTNO: consente TOUCH ai client di eseguire comandi senza influire suLRU/LFUdelle chiavi.
- Nuovo comando CLUSTER MYSHARDID che restituisce lo Shard ID del nodo per raggruppare logicamente i nodi in modalità cluster in base alla replica.
- Ottimizzazioni delle prestazioni e della memoria per vari tipi di dati.

Ecco le modifiche comportamentali potenzialmente negative tra Valkey 7.2 e Redis OSS 7.1 (o 7.0):

- Quando si chiama PUBLISH un RESP3 cliente che è anche iscritto allo stesso canale, l'ordine viene modificato e la risposta viene inviata prima del messaggio pubblicato.
- Il tracciamento degli script lato client ora tiene traccia delle chiavi lette dallo script, anziché delle chiavi dichiarate dal chiamante di/. EVAL FCALL
- Il freeze time sampling avviene durante l'esecuzione dei comandi e negli script.
- Quando un comando bloccato viene sbloccato, controlli comeACL, e altri vengono OOM rivalutati.
- ACLil testo del messaggio di errore e i codici di errore sono unificati.
- Un comando stream bloccato rilasciato quando la chiave non esiste più contiene un codice di errore diverso (- NOGROUP o - WRONGTYPE anziché -UNBLOCKED).
- Le statistiche del comando vengono aggiornate per i comandi bloccati solo quando il comando viene effettivamente eseguito.
- L'archiviazione interna degli ACL utenti non rimuove più le regole ridondanti di comando e categoria. Ciò può modificare il modo in cui tali regole vengono visualizzate come parte di ACLSAVE, ACL GETUSER e. ACL LIST
- Tutte le connessioni client create per l'utilizzo in modalità di replica TLS basata, SNI se possibile.
- XINFOSTREAM: Il campo di risposta in tempo di visualizzazione ora indica l'ultimo tentativo di interazione anziché l'ultima interazione riuscita. Il nuovo campo di risposta in tempo attivo ora indica l'ultima interazione riuscita.

- XREADGROUP e X [AUTO] CLAIM creano il consumatore indipendentemente dal fatto che sia stato in grado di eseguire alcune letture/affermazioni. [TBD- qual è il «problema» qui?]
- ACL l'utente predefinito appena creato imposta il flag `sanitize-payload in/`. `ACL LIST GETUSER`
- Il HELLO comando non influisce sullo stato del client a meno che non abbia esito positivo.
- NAN le risposte sono normalizzate in un singolo tipo nan, in modo simile al comportamento corrente di inf.

[Per ulteriori informazioni su Valkey, vedere Valkey](#)

[Per ulteriori informazioni sulla versione Valkey 7.2, consulta le note di rilascio di Redis OSS 7.2.4 \(Valkey 7.2 include tutte le modifiche rispetto a Redis OSS fino alla versione 7.2.4\) e le note di rilascio di Valkey 7.2 su Valkey su GitHub](#)

## Versioni Redis supportate OSS

ElastiCache Le cache serverless e le cache progettate internamente supportano tutte le versioni OSS Redis 7.1 e precedenti.

- [ElastiCache \(Valkey\) versione 7.2.6](#)
- [ElastiCache \(Redis\) versione 7.1 \(migliorataOSS\)](#)

I ElastiCache cluster progettati autonomamente supportano le seguenti versioni di Valkey e Redis: OSS

- [ElastiCache \(Redis\) versione 7.1 \(migliorataOSS\)](#)
- [ElastiCache \(Redis\) versione 7.0 \(migliorataOSS\)](#)
- [ElastiCache \(Redis\) versione 6.2 \(migliorataOSS\)](#)
- [ElastiCache \(Redis\) versione 6.0 \(migliorataOSS\)](#)
- [ElastiCache \(Redis\) versione 5.0.6 \(migliorataOSS\)](#)
- [ElastiCache \(RedisOSS\) versione 5.0.5 \(deprecata, usa la versione 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versione 5.0.4 \(deprecata, usa la versione 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versione 5.0.3 \(deprecata, usa la versione 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versione 5.0.0 \(obsoleta, usa la versione 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) versione 4.0.10 \(migliorata\)](#)
- [Versioni precedenti di End of Life \(EOL\) \(3.x\)](#)



- [Versioni precedenti di End of Life \(EOL\) \(2.x\)](#)

## ElastiCache (Redis) versione 7.1 (migliorataOSS)

Questa versione contiene miglioramenti delle prestazioni che consentono ai carichi di lavoro di aumentare il throughput 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 avanzati che ora non solo leggono l'input del client, ma lo analizzano anche nel formato di comando binario Redis. OSS L'input viene quindi inoltrato al thread principale per l'esecuzione, fornendo così un aumento delle prestazioni. Pattern di accesso alla memoria Redis OSS 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 utilizzano sistemi basati ElastiCache su Graviton3 R7g.4xlarge o versioni successive, i clienti possono raggiungere oltre 1 milione di richieste al secondo per nodo. Con i miglioramenti delle prestazioni di ElastiCache (RedisOSS) v7.1, i clienti possono ottenere fino al 100% in più di throughput e una latenza P99 inferiore del 50% rispetto a (Redis) v7.0. ElastiCache OSS Questi miglioramenti sono abilitati su nodi di dimensioni con almeno 8 core fisici (su Graviton e 2xlarge su x86), indipendentemente dal tipo e non richiedono modifiche 4xlarge al client. CPU

### Note

ElastiCache la versione 7.1 è compatibile con Redis v7.0. OSS

## ElastiCache (Redis) versione 7.0 (migliorataOSS)

ElastiCache (RedisOSS) 7.0 aggiunge una serie di miglioramenti e supporto per nuove funzionalità:

- [Funzioni](#): ElastiCache (RedisOSS) 7 aggiunge il supporto per Redis OSS Functions e offre un'esperienza gestita che consente agli sviluppatori di eseguire [LUAScript](#) con logica applicativa archiviata nel ElastiCache cluster, senza richiedere ai client di inviare nuovamente gli script al server ad ogni connessione.
- [ACLmiglioramenti](#): Valkey e Redis OSS 7 aggiungono il supporto per la prossima versione di Access Control Lists (). ACLs I client possono ora specificare più set di autorizzazioni su chiavi o spazi chiave specifici in Valkey e Redis. OSS

- [Sharded Pub/Sub](#): ElastiCache con Valkey e Redis OSS 7 aggiunge il supporto per l'esecuzione della funzionalità Pub/Sub in modo frammentato durante l'esecuzione in modalità Cluster Enabled (). ElastiCache CME Le funzionalità Pub/Sub consentono agli editori di inviare messaggi a qualsiasi numero di abbonati su un canale. I canali sono associati a uno shard nel ElastiCache cluster, eliminando la necessità di propagare le informazioni del canale tra gli shard, con conseguente migliore scalabilità.
- Multiplexing I/O migliorato: ElastiCache con Valkey e Redis OSS 7 introduce il multiplexing I/O avanzato, che offre un throughput maggiore e una latenza ridotta per carichi di lavoro ad alto throughput con molte connessioni client simultanee a un cluster. ElastiCache Ad esempio, quando si utilizza un cluster di nodi r6g.xlarge e si eseguono 5200 client simultanei, è possibile ottenere un aumento del throughput fino al 72% (operazioni di lettura e scrittura al secondo) e una riduzione della latenza P99 fino al 71%, ElastiCache rispetto alla OSS versione 6 di Redis.

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

## ElastiCache (Redis) versione 6.2 (migliorataOSS)

ElastiCache (RedisOSS) 6.2 include miglioramenti delle prestazioni per i cluster TLS abilitati che utilizzano tipi di nodi x86 con 8 vCPUs o più o tipi di nodi Graviton2 con 4 o più. vCPUs Questi miglioramenti migliorano la velocità effettiva e riducono i tempi di creazione della connessione del client affidando la crittografia ad altri. vCPUs Con Redis OSS 6.2, puoi anche gestire l'accesso ai canali Pub/Sub con le regole Access Control List (). ACL

Con questa versione, introduciamo anche il supporto per il tiering dei dati sui nodi del cluster contenenti dati collegati localmente. NVMe SSD Per ulteriori informazioni, consulta [Suddivisione dei dati su più livelli in ElastiCache](#).

La versione 6.2.6 OSS del motore Redis introduce anche il supporto per il formato nativo di JavaScript Object Notation (JSON), un modo semplice e senza schemi per codificare set di dati complessi all'interno dei cluster Redis. OSS Con il JSON supporto, puoi sfruttare le prestazioni e Redis per le applicazioni che operano su. OSS APIs JSON Per ulteriori informazioni, consulta [Guida introduttiva](#). JSON Sono incluse anche le metriche JSON relative a `JsonBasedCmds` e `JsonBasedCmdsLatency`, che vengono incorporate CloudWatch per monitorare l'utilizzo di questo tipo di dati. Per ulteriori informazioni, consulta [Metriche per Valkey e Redis OSS](#).

Si specifica la versione del motore utilizzando 6.2. ElastiCache (RedisOSS) richiamerà automaticamente la versione patch preferita di Redis OSS 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 la versione patch preferita attualmente disponibile di Redis OSS 6.2 al momento della creazione/modifica. Specificando la versione del motore 6.x in si API otterrà l'ultima versione secondaria di Redis 6. OSS

Per i cluster 6.0 esistenti, è possibile attivare il successivo aggiornamento automatico della versione secondaria impostando il `AutoMinorVersionUpgrade` parametro su `yes` in `CreateCacheCluster`, `ModifyCacheCluster` o `CreateReplicationGroup` `ModifyReplicationGroup` APIs ElastiCache (RedisOSS) aggiornerà la versione secondaria dei cluster 6.0 esistenti alla 6.2 utilizzando aggiornamenti self-service. Per ulteriori informazioni, consulta [Aggiornamenti self-service in Amazon ElastiCache](#).

Quando si chiama `DescribeCacheEngineVersions` API, il valore del `EngineVersion` parametro verrà impostato su 6.2 e nel campo verrà restituita la versione effettiva del motore con la versione della `CacheEngineVersionDescription` patch.

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

ElastiCache (Redis) versione 6.0 (migliorataOSS)

Amazon ElastiCache (RedisOSS) introduce la nuova versione del OSS motore Redis, che include [l'autenticazione degli utenti con Role Based Access Control, la memorizzazione](#) nella cache lato client e significativi miglioramenti operativi.

A partire da Redis OSS 6.0, ElastiCache (RedisOSS) offrirà un'unica versione per ogni versione secondaria di Redis, anziché offrire più versioni patch. OSS ElastiCache (RedisOSS) gestirà automaticamente la versione patch dei cluster di cache in esecuzione, garantendo prestazioni migliorate e maggiore sicurezza.

Puoi anche attivare il successivo aggiornamento automatico della versione secondaria impostando il `AutoMinorVersionUpgrade` parametro su `yes` e ElastiCache (RedisOSS) 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 (RedisOSS) richiamerà automaticamente la versione patch preferita di Redis OSS 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 la versione patch preferita attualmente disponibile di Redis OSS 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 DescribeCacheEngineVersions API, il valore del EngineVersion parametro verrà impostato su 6.0 e nel campo verrà restituita la versione effettiva del motore con la versione della patch. CacheEngineVersionDescription

Per ulteriori informazioni sulla versione Redis OSS 6.0, consulta le note di rilascio di [Redis OSS 6.0](#) su Redis on. OSS GitHub

### ElastiCache (Redis) versione 5.0.6 (migliorataOSS)

Amazon ElastiCache (RedisOSS) introduce la prossima versione del OSS 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.
- Capacità di rinominare i comandi: un parametro chiamato `rename-commands` che consente di rinominare OSS comandi Redis potenzialmente pericolosi o costosi che potrebbero causare la perdita accidentale di dati, come o. FLUSHALL FLUSHDB È simile alla configurazione `rename-command` in Redis open source. OSS 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 ai parametri di Redis 5.0.3 OSS](#).

- Redis OSS Streams: Questo modello una struttura di dati di registro che consente ai produttori di aggiungere 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 Streams](#).
- Support per una famiglia di comandi di flusso come XADD, XRANGE e XREAD. Per ulteriori informazioni, consulta [Streams Commands](#).
- Alcuni parametri nuovi e ridenominati. Per ulteriori informazioni, consulta [Modifiche ai parametri di Redis 5.0.0 OSS](#).
- Una nuova OSS metrica Redis, `StreamBasedCmds`
- Tempo di snapshot leggermente più rapido per i nodi Redis. OSS

 Important

ElastiCache (RedisOSS) ha ripristinato due correzioni di bug critici dalla versione [open source 5.0.1 di Redis. OSS](#) Sono elencati qui di seguito:

- RESTORErisposta non corrispondente quando alcune 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 supporto ElastiCache (RedisOSS) per la versione 5.0.0 OSS del motore Redis e vengono utilizzate negli aggiornamenti delle versioni future.

Per ulteriori informazioni, consulta le note di rilascio di [Redis OSS 5.0.6 su Redis on. OSS GitHub](#)

ElastiCache (RedisOSS) versione 5.0.5 (deprecata, usa la versione 5.0.6)

Amazon ElastiCache (RedisOSS) introduce la prossima versione del motore RedisOSS;. Include modifiche alla configurazione online per ElastiCache (RedisOSS) dei cluster di failover automatico durante tutte le operazioni pianificate. Ora puoi scalare il cluster, aggiornare la versione del OSS

motore Redis e applicare patch e aggiornamenti di manutenzione mentre il cluster rimane online e continua a soddisfare le richieste in arrivo. Include anche correzioni di bug.

Per ulteriori informazioni, consulta le note di [rilascio di Redis OSS 5.0.5](#) su Redis on. OSS GitHub ElastiCache (RedisOSS) versione 5.0.4 (deprecata, usa la versione 5.0.6)

Amazon ElastiCache (RedisOSS) introduce la prossima versione del OSS motore Redis supportato 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 OSS 5.0.4](#) su Redis on. OSS GitHub ElastiCache (RedisOSS) versione 5.0.3 (deprecata, usa la versione 5.0.6)

Amazon ElastiCache (RedisOSS) introduce la prossima versione del OSS motore Redis supportato da Amazon ElastiCache che include correzioni di bug.

ElastiCache (RedisOSS) versione 5.0.0 (obsoleta, usa la versione 5.0.6)

Amazon ElastiCache (RedisOSS) introduce la prossima versione principale del OSS motore Redis supportato da Amazon. ElastiCache ElastiCache (RedisOSS) 5.0.0 supporta i seguenti miglioramenti:

- Redis OSS Streams: modella una struttura di dati di log che consente ai produttori di aggiungere 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 Streams](#).
- Support per una famiglia di comandi di flusso come XADD, XRANGE e XREAD. Per ulteriori informazioni, consulta [Streams](#) Commands.
- Alcuni parametri nuovi e ridenominati. Per ulteriori informazioni, consulta [Modifiche ai parametri di Redis 5.0.0 OSS](#).

- Una nuova OSS metrica Redis, `StreamBasedCmds`
- Tempo di snapshot leggermente più rapido per i nodi Redis. OSS

## ElastiCache (RedisOSS) versione 4.0.10 (migliorata)

Amazon ElastiCache (RedisOSS) introduce la prossima versione principale del OSS motore Redis supportato da Amazon. ElastiCache ElastiCache (RedisOSS) 4.0.10 supporta i seguenti miglioramenti:

- Ridimensionamento e crittografia dei cluster online in un'unica ElastiCache versione (Redis). OSS  
Per ulteriori informazioni, consulta quanto segue:
  - [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)
  - [Resharding online per Valkey o Redis OSS \(modalità cluster abilitata\)](#)
  - [Sicurezza dei dati in Amazon ElastiCache](#)
- Alcuni nuovi parametri. Per ulteriori informazioni, consulta [Modifiche ai parametri Redis OSS 4.0.10](#).
- Support per la famiglia di comandi di memoria, come MEMORY. Per ulteriori informazioni, consulta [Comandi](#) (ricerca attiva). 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 lavaggi ed eliminazioni asincroni. ElastiCache (RedisOSS) supporta comandi come UNLINK FLUSHDB e da eseguire in un thread FLUSHALL diverso dal thread principale. Questo consente di migliorare i tempi di risposta e le prestazioni per le applicazioni liberando la memoria in modo asincrono.
- Una nuova OSS metrica Redis, `ActiveDefragHits` Per ulteriori informazioni, consulta [Metrics for Redis. OSS](#)

Gli utenti di Valkey o Redis OSS (modalità cluster disabilitata) che eseguono la OSS versione 3.2.10 di Redis possono utilizzare la console per aggiornare i propri cluster tramite l'aggiornamento online.



## Confronto tra il ridimensionamento dei cluster ElastiCache (RedisOSS) e il supporto per la crittografia

| 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.

\*\* Necessario per applicazioni Fed RAMP HIPAA e PCI DSS conformi. Per ulteriori informazioni, consulta [Convalida della conformità per Amazon ElastiCache](#).

## Versioni precedenti di End of Life (EOL) (3.x)

## ElastiCache (RedisOSS) versione 3.2.10 (migliorata)

Amazon ElastiCache (RedisOSS) introduce la prossima versione principale del OSS motore Redis supportato da Amazon. ElastiCache (RedisOSS) 3.2.10 introduce il ridimensionamento online del cluster per aggiungere o rimuovere frammenti dal cluster mentre continua a soddisfare le richieste di I/O in entrata. ElastiCache (RedisOSS) 3.2.10 gli utenti dispongono di tutte le funzionalità delle versioni precedenti di Redis ad eccezione della possibilità di crittografare i propri dati. OSS Questa caratteristica è attualmente disponibile nella versione 3.2.6.

## Confronto tra le versioni ElastiCache (RedisOSS) 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     |

| Funzionalità                                                                                                                                                                    | 3.2.6 | 3.2.10 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| * Aggiunta, rimozione e ribilanciamento delle partizioni.                                                                                                                       |       |        |
| ** Obbligatorio per le applicazioni Fed RAMP e HIPAA conformi. PCI DSS Per ulteriori informazioni, consulta <a href="#">Convalida della conformità per Amazon ElastiCache</a> . |       |        |

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Resharding online per Valkey o Redis OSS \(modalità cluster abilitata\)](#)
- [Ridimensionamento di cluster online](#)

#### ElastiCache (RedisOSS) versione 3.2.6 (migliorata)

Amazon ElastiCache (RedisOSS) introduce la prossima versione principale del OSS motore Redis supportato da Amazon. ElastiCache ElastiCache (RedisOSS) 3.2.6 gli utenti dispongono di tutte le funzionalità delle OSS versioni precedenti di Redis oltre alla possibilità di crittografare i propri dati. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [ElastiCache crittografia in transito \(\) TLS](#)
- [Crittografia At-Rest in ElastiCache](#)
- [Convalida della conformità per Amazon ElastiCache](#)

#### ElastiCache (Redis) versione 3.2.4 (migliorataOSS)

La versione 3.2.4 di Amazon ElastiCache (RedisOSS) introduce la prossima versione principale del motore Redis OSS supportata da Amazon. ElastiCache ElastiCache (RedisOSS) 3.2.4 hanno a disposizione tutte le funzionalità delle OSS versioni precedenti di Redis, oltre alla possibilità di eseguire in modalità cluster o non cluster. La tabella seguente funge da riepilogo.

Confronto tra la modalità non cluster di Redis OSS 3.2.4 e la modalità cluster

| Funzionalità               | Modalità non cluster | Modalità cluster |
|----------------------------|----------------------|------------------|
| Partizionamento dei dati   | No                   | Sì               |
| Indicizzazione geospaziale | Sì                   | Sì               |

| Funzionalità                   | Modalità non cluster             | Modalità cluster                            |
|--------------------------------|----------------------------------|---------------------------------------------|
| 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 supporto della replica per ciascun gruppo di nodi.
- Indicizzazione geospaziale: Redis OSS 3.2.4 introduce il supporto per l'indicizzazione geospaziale tramite sei comandi. GEO [Per ulteriori informazioni, consulta la documentazione del comando Redis OSS GEO \\* Comandi: nella pagina Comandi Valkey \(filtrata per\).](#) GEO GEO

[Per informazioni sulle funzionalità aggiuntive di Redis OSS 3, consultate le note di rilascio di Redis OSS 3.2 e le note di rilascio di Redis 3.0. OSS](#)

Valkey o Redis attualmente ElastiCache gestiti OSS (modalità cluster abilitata) non supportano le seguenti funzionalità di Redis 3.2: OSS

- Migrazione di repliche
- Ribilanciamento di cluster
- Debugger Lua

ElastiCache disabilita i seguenti comandi di gestione Redis 3.2OSS:

- `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 OSS 3.2.4, consulta [Modifiche ai parametri di Redis 3.2.4 OSS](#)

Versioni precedenti di End of Life (EOL) (2.x)

ElastiCache (RedisOSS) versione 2.8.24 (migliorata)

OSSI miglioramenti di Redis aggiunti dalla versione 2.8.23 includono correzioni di bug e registrazione di indirizzi di accesso errati alla memoria. [Per ulteriori informazioni, consulta le note di rilascio di Redis 2.8. OSS](#)

ElastiCache (RedisOSS) versione 2.8.23 (migliorata)

OSSI miglioramenti di Redis aggiunti dalla versione 2.8.22 includono correzioni di bug. Per ulteriori informazioni, consulta le note di rilascio di [Redis 2.8 OSS](#). 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 OSS 2.8.23, consulta [Parametri aggiunti per Redis OSS 2.8.23 \(migliorato\)](#) la Guida per l'utente. ElastiCache

ElastiCache (RedisOSS) versione 2.8.22 (migliorata)

OSSI miglioramenti di Redis aggiunti dalla versione 2.8.21 includono quanto segue:

- 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 nuove CloudWatch metriche.
  - `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 CloudWatch Metrics](#).

- Una serie di correzioni di bug critici nel comportamento di replicaPSYNC. Per ulteriori informazioni, consulta le note di rilascio di [Redis OSS 2.8](#).
- Per mantenere prestazioni di replica migliorate nei gruppi di replica Multi-AZ e per una maggiore stabilità del cluster, le non ElastiCache repliche 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 OSS configurazione Redis `appendonly` e `appendfsync` sono supportate nella versione OSS Redis 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. È più probabile che tali situazioni si verifichino per i clienti. PUBSUB

### ElastiCache (RedisOSS) versione 2.8.21

OSSI miglioramenti di Redis aggiunti dalla versione 2.8.19 includono una serie di correzioni di bug. Per ulteriori informazioni, consulta le note di rilascio di [Redis 2.8 OSS](#).

### ElastiCache (RedisOSS) versione 2.8.19

OSSI miglioramenti di Redis aggiunti dalla versione 2.8.6 includono quanto segue:

- Support per HyperLogLog. Per ulteriori informazioni, consulta la [OSSnuova struttura dati di 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 a un nodo primario di inviare dati non aggiornati ai nodi di replica, il master SYNC fallisce se viene interrotto un processo secondario save (bgsave) in background.
- Support per la HyperLogLogBasedCommands CloudWatchmetrica. Per ulteriori informazioni, consulta [Metriche per Valkey e Redis OSS](#).

## ElastiCache (RedisOSS) versione 2.8.6

OSSI miglioramenti di Redis aggiunti dalla versione 2.6.13 includono quanto segue:

- 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 (Redis) versione 2.6.13 OSS

La OSS versione 2.6.13 di Redis era la versione iniziale di Redis supportata OSS da Amazon ElastiCache (Redis). OSS Multi-AZ non è supportato su Redis 2.6.13. OSS

## Pianificazione di fine vita delle versioni Redis OSS



Questa sezione definisce le date di fine vita (EOL) per le versioni principali precedenti man mano che vengono annunciate. Ciò consente di prendere decisioni future sulla versione e sugli aggiornamenti.

### Note

ElastiCache Le versioni delle patch (RedisOSS) dalla 5.0.0 alla 5.0.5 sono obsolete. Usa versione 5.0.6 o successiva.

La tabella seguente riassume ogni versione e la EOL data annunciata, nonché la versione di destinazione dell'aggiornamento consigliata.

### Passato EOL

| Versione | Versioni secondarie di origine                                | Destinazione aggiornamento consigliata                                                                                                                                                                                                                                                                                                                                        | EOL Data        |
|----------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 3        | 3.2.4, 3.2.6 e 3.2.10                                         | Versione 6.2 o successiva<br><br><div data-bbox="613 730 1040 1192" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>Per le regioni US-ISO-EAST US-ISO-WEST -1, -1 e US-ISOB-EAST -1, consigliamo la versione 5.0.6 o successiva.</p> </div>   | 31 luglio 2023  |
| 2        | 2.8.24, 2.8.23, 2.8.22, 2.8.21, 2.8.19, 2.8.12, 2.8.6, 2.6.13 | Versione 6.2 o successiva<br><br><div data-bbox="613 1304 1040 1724" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>Per le US-ISO-EAST regioni -1 US-ISO-WEST, -1 US-ISOB-EAST e -1, consigliamo la versione 5.0.6 o superiore.</p> </div> | 13 gennaio 2023 |

## Versioni supportate ElastiCache (Memcached)

ElastiCache supporta le seguenti versioni di Memcached e l'aggiornamento a versioni più recenti. Per effettuare l'aggiornamento a una versione più recente, occorre prestare particolare attenzione alle condizioni che, se non soddisfatte, ne possono causare la mancata esecuzione.

ElastiCache per le versioni Memcached

- [Memcached versione 1.6.22](#)
- [Memcached versione 1.6.17](#)
- [Memcached versione 1.6.12](#)
- [Memcached versione 1.6.6](#)
- [Memcached versione 1.5.16](#)
- [Memcached versione 1.5.10](#)
- [Memcached versione 1.4.34](#)
- [Memcached versione 1.4.33](#)
- [Memcached versione 1.4.24](#)
- [Memcached versione 1.4.14](#)
- [Memcached versione 1.4.5](#)

Memcached versione 1.6.22

ElastiCache (Memcached) aggiunge il supporto per la versione 1.6.22 di Memcached. Non include nuove funzionalità, ma include correzioni di bug e aggiornamenti cumulativi da [Memcached 1.6.18](#).

[Per ulteriori informazioni, vedere 1622 su Memcached on. ReleaseNotes](#) GitHub

Memcached versione 1.6.17

ElastiCache (Memcached) aggiunge il supporto per la versione 1.6.17 di Memcached. Non include nuove funzionalità, ma include correzioni di bug e aggiornamenti cumulativi da [Memcached 1.6.17](#).

[Per ulteriori informazioni, vedere 1617 su Memcached on. ReleaseNotes](#) GitHub

Memcached versione 1.6.12

ElastiCache (Memcached) aggiunge il supporto per la versione 1.6.12 di Memcached e la crittografia in transito. Inoltre, include correzioni di bug e aggiornamenti cumulativi da [Memcached 1.6.6](#).



[Per ulteriori informazioni, vedere 1612 su Memcached on. ReleaseNotes GitHub](#)

### Memcached versione 1.6.6

ElastiCache (Memcached) aggiunge il supporto per la versione 1.6.6 di Memcached. [Non include nuove funzionalità, ma include correzioni di bug e aggiornamenti cumulativi da Memcached 1.5.16.](#)

ElastiCache [\(Memcached\) non include il supporto per Extstore.](#)

Per ulteriori informazioni, vedere [ReleaseNotes166](#) su Memcached on. GitHub

### Memcached versione 1.5.16

ElastiCache for Memcached aggiunge il supporto per la versione 1.5.16 di Memcached. Non include nuove caratteristiche, ma include correzioni di bug e aggiornamenti cumulativi da [Memcached 1.5.14](#) e [Memcached 1.5.15](#).

Per ulteriori informazioni, vedere le note di rilascio di Memcached 1.5.16 su [Memcached](#) su. GitHub

### Memcached versione 1.5.10

ElastiCache per Memcached la versione 1.5.10 supporta le seguenti funzionalità di Memcached:

- Ribilanciamento automatizzato dell'allocazione.
- Ricerche più rapide nella tabella hash con l'algoritmo murmur3.
- LRUAlgoritmo segmentato.
- LRUcrawler per recuperare la memoria in background.
- `--enable-seccomp`: un'opzione della fase di compilazione.

Introduce anche i parametri `no_modern` e `inline_ascii_resp`. Per ulteriori informazioni, consulta [Modifiche dei parametri di Memcached 1.5.10](#).

I miglioramenti di Memcached aggiunti dopo la versione 1.4.34 di Memcached includono quanto segue ElastiCache :

- Correzioni cumulative, come multiget, -2017-9951 e limiti di indicizzazione per. ASCII CVE metadumper
- Migliore gestione delle connessioni mediante la chiusura delle connessioni una volta raggiunto il limite di connessioni.
- Gestione migliorata delle dimensioni per elementi di dimensioni maggiori di 1 MB.

- Migliori prestazioni e miglioramenti del sovraccarico di memoria mediante la riduzione di qualche byte dei requisiti di memoria per voce.

[Per ulteriori informazioni, vedere le note di rilascio di Memcached 1.5.10 su Memcached on. GitHub](#)

#### Memcached versione 1.4.34

ElastiCache per la versione 1.4.34 di Memcached non aggiunge nuove funzionalità alla versione 1.4.33. La 1.4.34 è una versione di correzione dei bug più estesa del solito.

Per ulteriori informazioni, vedere le note di rilascio di Memcached 1.4.34 su [Memcached](#) su. GitHub

#### Memcached versione 1.4.33

Tra i miglioramenti di Memcached applicati a partire dalla versione 1.4.24 sono inclusi:

- Capacità di esecuzione del dump dei metadati afferenti a una classe, un elenco di classi o tutte le classi di slab. Per ulteriori informazioni, consulta le [note di rilascio di Memcached 1.4.31](#).
- Supporto migliorato per elementi di grandi dimensioni, superiori al megabyte di default. Per ulteriori informazioni, consulta le [note di rilascio di Memcached 1.4.29](#).
- Capacità di stabilire quanto tempo un client può rimanere inattivo prima che ne venga richiesta la chiusura.

Capacità di aumentare dinamicamente la quantità di memoria disponibile per Memcached senza dover riavviare il cluster. Per ulteriori informazioni, consulta le [note di rilascio di Memcached 1.4.27](#).

- Registrazione di `fetchers`, `mutations` ed `evictions` ora supportata. Per ulteriori informazioni, consulta le [note di rilascio di Memcached 1.4.26](#).
- Memoria liberata recuperabile in un pool globale e riassegnabile a nuove classi di slab. Per ulteriori informazioni, consulta le [note di rilascio di Memcached 1.4.25](#).
- Risoluzione di vari bug.
- Nuovi comandi e parametri. Per un elenco, consulta [Parametri di Memcached 1.4.33 aggiunti](#).

#### Memcached versione 1.4.24

Tra i miglioramenti di Memcached applicati a partire dalla versione 1.4.14 sono inclusi:

- Gestione utilizzata meno di recente (LRU) utilizzando un processo in background.
- Possibilità di utilizzo di jenkins o murmur3 come algoritmo hash.

- Nuovi comandi e parametri. Per un elenco, consulta [Parametri di Memcached 1.4.24 aggiunti](#).
- Risoluzione di vari bug.

### Memcached versione 1.4.14

Tra i miglioramenti di Memcached applicati a partire dalla versione 1.4.5 sono inclusi:

- Capacità potenziata di ribilanciamento dell'allocazione.
- Miglioramenti nelle prestazioni e nella scalabilità.
- Introduzione del comando touch, utile per aggiornare la scadenza di un elemento esistente senza il recupero.
- Auto discovery - consente ai programmi client di identificare automaticamente tutti i nodi cache in un cluster e di avviare e mantenere le connessioni a tutti questi nodi.

### Memcached versione 1.4.5

La versione 1.4.5 di Memcached era il motore e la versione iniziale supportati da Amazon ElastiCache (Memcached).

## Principali differenze di comportamento e compatibilità della versione con Valkey

Valkey 7.2.6 presenta differenze di compatibilità simili con le versioni precedenti di Redis 7.2.5. OSS Per la versione più recente supportata di Valkey, vedi. [Motori e versioni supportati](#)

## Principali differenze di comportamento e compatibilità delle versioni con Redis OSS

### 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.

È possibile eseguire l'aggiornamento direttamente dalla versione Redis corrente all'ultima OSS versione Redis disponibile, senza la necessità di aggiornamenti OSS sequenziali. Ad esempio, puoi eseguire l'aggiornamento direttamente dalla versione 3.0 di Redis alla OSS versione 7.0.

OSSLe versioni Redis sono identificate con una versione semantica che comprende un MAJOR componente, e. MINOR PATCH Ad esempio, in Redis OSS 4.0.10, la versione principale è 4, la versione secondaria 0 e la versione patch è 10. Questi valori generalmente vengono incrementati in base alle convenzioni seguenti:

- MAJORLe versioni riguardano modifiche incompatibili API
- MINORLe versioni riguardano nuove funzionalità aggiunte in modo retrocompatibile
- PATCHLe versioni riguardano correzioni di bug compatibili con le versioni precedenti e modifiche non funzionali

Consigliamo di rimanere sempre aggiornati all'ultima versione della patch entro un determinato periodo. MAJOR MINORversione per avere gli ultimi miglioramenti in termini di prestazioni e stabilità. A partire da Redis OSS 6.0, ElastiCache (RedisOSS) offrirà un'unica versione per ogni versione OSS minore di Redis, anziché offrire più versioni di patch. ElastiCache (RedisOSS) gestirà

automaticamente la versione patch 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. Poiché ElastiCache estende la disponibilità a una nuova AWS regione, ElastiCache (RedisOSS) supporta le due più recenti. MAJOR MINOR versioni in quel momento per la nuova regione. Ad esempio, se viene lanciata una nuova AWS regione e la versione più recente MAJOR. MINOR ElastiCache Le versioni (RedisOSS) sono 7.0 e 6.2, ElastiCache (RedisOSS) supporterà le versioni 7.0 e 6.2 nella nuova regione. AWS Come più recente. MAJOR MINOR Le versioni di ElastiCache (RedisOSS) sono state rilasciate, ElastiCache continueranno ad aggiungere il supporto per le versioni appena rilasciate ElastiCache (RedisOSS). Per ulteriori informazioni sulla scelta delle aree per cui ElastiCache, vedi [Scelta delle regioni e delle zone di disponibilità](#).

Quando esegui un aggiornamento che include versioni principali o secondarie, prendi in considerazione il seguente elenco, che include il comportamento e le modifiche retrocompatibili rilasciate con Redis OSS nel tempo.

## Comportamento di Redis OSS 7.0 e modifiche incompatibili con le versioni precedenti

Per un elenco completo delle modifiche, consulta le note di rilascio di [Redis 7.0 OSS](#).

- `SCRIPT LOAD` e `SCRIPT FLUSH` non sono più propagati alle repliche. [Se hai bisogno di una certa durabilità per gli script, ti consigliamo di prendere in considerazione l'utilizzo delle funzioni Redis OSS](#)
- 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 ACL categorie per `SELECT`, `WAIT`, `ROLE`, `LASTSAVE` `READONLY` `READWRITE`, e `ASKING` sono state modificate.
- 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.

## Comportamento di Redis OSS 6.2 e modifiche incompatibili con le versioni precedenti

Per un elenco completo delle modifiche, consulta le note di rilascio di [Redis 6.2 OSS](#).

- I ACL flag dei LASTSAVE comandi TIME, ECHO ROLE, e 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.

- Durante l'aggiornamento da Redis OSS 6.0, l'ordine delle coppie chiave/valore restituite da una risposta della mappa a uno script lua viene modificato. Se i tuoi script utilizzano `redis.setresp()` o restituiscono una mappa (novità in Redis OSS 6.0), considera le implicazioni che lo script potrebbe non funzionare durante gli aggiornamenti.

## Comportamento di Redis OSS 6.0 e modifiche non compatibili con le versioni precedenti

Per un elenco completo delle modifiche, consulta le note di rilascio di [Redis 6.0 OSS](#).

- 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 `AutoMinorVersionUpgrade` parametro su `yes` e ElastiCache (RedisOSS) gestirà l'aggiornamento della versione secondaria 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

## Comportamento di Redis OSS 5.0 e modifiche non compatibili con le versioni precedenti

Per un elenco completo delle modifiche, consulta le note di rilascio di [Redis 5.0 OSS](#).

- 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 ripristinare il comportamento precedente, disponibile solo in ElastiCache (RedisOSS) 5.0.
- Se state eseguendo l'aggiornamento da Redis OSS 4.0, alcuni comandi negli LUA script restituiranno gli argomenti in un ordine diverso rispetto alle versioni precedenti. In Redis OSS 4.0, Redis OSS ordinava alcune risposte in modo lessografico per rendere le risposte deterministiche. Questo ordinamento non viene applicato quando gli script vengono replicati mediante effetti.
- In Redis OSS 5.0.3 e versioni successive, ElastiCache (RedisOSS) trasferirà parte del lavoro di I/O sui core in background su tipi di istanze con più di 4. VCPUs Ciò potrebbe modificare le caratteristiche prestazionali di Redis OSS e modificare i valori di alcune metriche. Per ulteriori informazioni, consulta [Quali parametri è opportuno monitorare?](#) per appurare se è necessario modificare le metriche.

## Comportamento di Redis OSS 4.0 e modifiche incompatibili con le versioni precedenti

Per un elenco completo delle modifiche, consulta le note di rilascio di [Redis 4.0 OSS](#).

- 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.

## Passato EOL

### Comportamento di Redis OSS 3.2 e modifiche incompatibili con le versioni precedenti

Per un elenco completo delle modifiche, consulta le note di rilascio di [Redis 3.2 OSS](#).

- Non esistono modifiche di compatibilità da richiamare per questa versione.

Per ulteriori informazioni, consulta [Pianificazione di fine vita delle versioni Redis OSS](#).

### Comportamento di Redis OSS 2.8 e modifiche retrocompatibili

Per un elenco completo delle modifiche, consulta le note di rilascio di [Redis 2.8 OSS](#).

- A partire da Redis OSS 2.8.22, Redis non è più supportato in (OSSAOFRedis). ElastiCache OSS È preferibile utilizzare MemoryDB quando i dati devono essere conservati in modo duraturo.
- A partire da Redis OSS 2.8.22, ElastiCache (RedisOSS) non supporta più il collegamento di repliche ai file primari ospitati all'interno. ElastiCache Durante l'aggiornamento, le repliche esterne verranno scollegate e non potranno ricollegarsi. Consigliamo di utilizzare la memorizzazione nella cache lato client, disponibile in Redis 6.0 come alternativa alle repliche esterne. OSS
- I comandi TTL e PTTL ora restituiscono -2 se la chiave non esiste e -1 se esiste ma non ha una scadenza associata. Redis OSS 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 di fine vita delle versioni Redis OSS](#).

## Risoluzione degli aggiornamenti del motore Valkey o Redis OSS bloccati

Come illustrato nella tabella seguente, l'operazione di aggiornamento del OSS motore Valkey o Redis è bloccata se è in corso un'operazione di scalabilità.

| 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 |

Per risolvere un aggiornamento del motore Redis bloccato OSS

- Esegui una di queste operazioni:
  - Pianifica l'operazione di aggiornamento OSS del motore Redis per la prossima finestra di manutenzione deselegionando la casella di controllo Applica immediatamente.

ConCLI, usa. `--no-apply-immediately` ConAPI, usa `ApplyImmediately=false`.



- Attendi la finestra di manutenzione successiva (o successiva) per eseguire l'operazione di aggiornamento OSS del motore Redis.
- Aggiungi l'operazione di OSS scalabilità Redis a questa modifica del cluster selezionando la casella di controllo Applica immediatamente.

ConCLI, usa `--apply-immediately`. ConAPI, usa `ApplyImmediately=true`.

Questo approccio annulla efficacemente l'aggiornamento del motore nella finestra di manutenzione successiva, poiché lo si esegue immediatamente.

## ElastiCache migliori pratiche e strategie di caching

Di seguito puoi trovare le best practice consigliate per Amazon ElastiCache. Queste best practice consentono di migliorare prestazioni e affidabilità della cache.

### Argomenti

- [Le migliori pratiche generali](#)
- [Comandi Valkey, Redis e Memcached supportati OSS e limitati](#)
- [OSSConfigurazione e limiti di Valkey e Redis](#)
- [IPv6esempi di client per Valkey, Redis OSS e Memcached](#)
- [Le migliori pratiche per i clienti \(Valkey e RedisOSS\)](#)
- [Le migliori pratiche per i clienti \(Memcached\)](#)
- [TLScluster dual stack ElastiCache abilitati](#)
- [Gestione della memoria riservata per Valkey e Redis OSS](#)
- [Le migliori pratiche per lavorare con cluster progettati autonomamente da Valkey e Redis OSS](#)
- [Strategie di caching per Memcached](#)

## Le migliori pratiche generali

Di seguito puoi trovare informazioni sulle migliori pratiche per l'utilizzo delle interfacce Valkey, Redis OSS e Memcached all'interno. ElastiCache

- Usa configurazioni abilitate per la modalità cluster: la modalità cluster abilitata consente alla cache di scalare orizzontalmente per ottenere uno storage e un throughput più elevati rispetto a una

configurazione disabilitata in modalità cluster. ElastiCache serverless è disponibile solo in una configurazione abilitata per la modalità cluster.

- Utilizza connessioni di lunga durata: la creazione di una nuova connessione è costosa e richiede tempo e CPU risorse dalla 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 sistemi ElastiCache serverless o hai predisposto 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 modalità ElastiCache serverless, la lettura dalla porta di replica (6380) indirizzerà le letture verso la zona di disponibilità locale del client, ove possibile, riducendo la latenza di recupero. Eseguie 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 sharded pub/sub: [quando usi Valkey o Redis OSS per supportare carichi di lavoro pub/sub con throughput elevato, ti consigliamo di utilizzare pub/sub sharded \(disponibile con Valkey e con Redis 7 o versioni successive\)](#). OSS 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 nei comandi pub/sub tradizionali senza server, internamente vengono utilizzati comandi pub/sub condivisi. ElastiCache

# Comandi Valkey, Redis e Memcached supportati OSS e limitati

## Comandi Valkey e Redis supportati OSS

### Comandi Valkey e Redis supportati OSS

I seguenti OSS comandi Valkey e Redis sono supportati dalle cache serverless. Oltre a questi comandi, sono supportati anche i [Comandi Valkey e Redis OSS 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 Valkey o Redis Cluster. OSS

[Ulteriori informazioni](#)

- READWRITE

Abilita le query di lettura/scrittura per una connessione a un nodo di replica Valkey o Redis Cluster. OSS

[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

Strette di mano con il server Valkey o Redis. OSS

[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](#)

## HyperLogLog Comandi

- PFADD

Aggiunge elementi a una HyperLogLog chiave. Crea la chiave, se non esiste.

[Ulteriori informazioni](#)

- PFCOUNT

Restituisce la cardinalità approssimativa degli insiemi osservati dalle HyperLogLog chiavi.

[Ulteriori informazioni](#)

- **PFMERGE**

Unisce uno o più HyperLogLog valori 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](#)



- RPU SH

Aggiunge uno o più elementi a un elenco. Crea la chiave, se non esiste.


[Ulteriori informazioni](#)

- RPU SHX

Aggiunge un elemento a un elenco solo quando l'elenco esiste.

[Ulteriori informazioni](#)

## Comandi Pub/Sub

 Note

PUBSUBi comandi utilizzano internamente shardedPUBSUB, quindi i nomi dei canali verranno misti.

- 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 sul lato server tramite digest. SHA1

[Ulteriori informazioni](#)

- EVALSHA\_RO

Esegue uno script Lua lato server di sola lettura tramite 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 ACL categorie 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 ACL regole di un utente.

[Ulteriori informazioni](#)

- ACL LIST

Scarica le regole valide in formato ACL file.

[Ulteriori informazioni](#)

- ACL USERS

Elenca tutti gli ACL utenti.

[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 grafica computerizzata e la versione Valkey o RedisOSS.

[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](#)

- BZPOP MAX

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](#)

- ZREMRANGEBYSORE

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](#)

- ZREVRANGEBYSORE

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 da uno stream all'interno di un intervallo di IDs

[Ulteriori informazioni](#)

- XREAD

Restituisce messaggi da più stream con IDs un numero maggiore di 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 da uno stream all'interno di un intervallo IDs 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 Valkey e Redis OSS con restrizioni

Per offrire un'esperienza di servizio gestito, ElastiCache limita l'accesso a determinati 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 getredis`
- `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`

## Comandi Memcached supportati

ElastiCache Serverless for Memcached supporta tutti i [comandi](#) memcached in memcached 1.6 open source ad eccezione dei seguenti:

- Le connessioni client richiedono TLS, di conseguenza, che il protocollo non è supportato. UDP
- Il protocollo binario non è supportato, in quanto è ufficialmente [obsoleto](#) in memcached 1.6.
- I comandi GET/GETS sono limitati a 16 KB per evitare potenziali attacchi DoS al server con il recupero di un numero elevato di chiavi.
- Il comando `flush_all` ritardato verrà rifiutato con `CLIENT_ERROR`.
- I comandi che configurano il motore o rivelano informazioni interne sullo stato o sui log del motore non sono supportati, ad esempio:
  - Per il comando `STATS` sono supportati solo `stats` e `stats reset`. Altre varianti restituiscono `ERROR`.
  - `lru / lru_crawler`- modifica delle impostazioni LRU e del LRU crawler
  - `watch`: controlla i log del server memcached.
  - `verbosity`: configura il livello di log del server.
  - `me`- il comando meta debug (`me`) non è supportato

## OSSConfigurazione e limiti di Valkey e Redis

I OSS motori Valkey e Redis forniscono ciascuno una serie di parametri di configurazione, alcuni dei quali sono modificabili in ElastiCache (RedisOSS) e altri non modificabili per fornire prestazioni e affidabilità stabili.

### Cache serverless

Per le cache serverless, i gruppi di parametri non vengono utilizzati e tutte le configurazioni di Valkey o Redis non sono modificabili. OSS Sono presenti i seguenti parametri Valkey o Redis: OSS

| Nome                                              | Informazioni             | Descrizione                                                                  |
|---------------------------------------------------|--------------------------|------------------------------------------------------------------------------|
| <code>acl-pubsub</code><br><code>b-default</code> | <code>allchannels</code> | Autorizzazioni predefinite del canale pubsub per ACL gli utenti nella cache. |

| Nome                                             | Informazioni                                                 | Descrizione                                                                                                                                                                                                                                                                     |
|--------------------------------------------------|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>client-output-buffer-limit</code>          | <code>normal 0 0 0</code><br><code>pubsub 32mb 8mb 60</code> | I client normali non hanno limiti di buffer. PUB/SUBi client verranno disconnessi se violano il backlog di 32 MiB o violano il backlog di 8 MiB per 60.                                                                                                                         |
| <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 partizionare in modo trasparente i dati su più partizioni 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> . |



| Nome                                | Informazioni                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>lua-time-limit</code>         | 5000                                             | <p>Il tempo massimo di esecuzione per uno script Lua, in millisecondi, prima che venga eseguita un'azione per interrompere lo script. ElastiCache</p> <p>Se <code>lua-time-limit</code> viene superato, tutti i OSS comandi Valkey o Redis possono restituire un errore nel formato <code>____-BUSY</code>. Poiché questo stato può causare interferenze con molte OSS operazioni essenziali di Valkey o Redis, emetterà prima un comando. ElastiCache <code>SCRIPTKILL</code>. Se ciò non riesce, ElastiCache riavvierà forzatamente Valkey o Redis. OSS</p> |
| <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 TTL set vengono eliminati in base alla stima <code>least-recently-used</code> (LRU) quando viene raggiunto il limite di memoria della cache.                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>notify-keyspace-events</code> | Stringa vuota                                    | Gli eventi del keyspace non sono attualmente supportati nelle cache serverless.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>port</code>                   | Porta principale: 6379<br>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 <code>READONLY</code> .                                                                                                                                                                                                                                                                                              |

| Nome                   | Informazioni | Descrizione                                                                                                                                                                                       |
|------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| proto-max<br>-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 per una singola chiave Valkey o Redis OSS o il nome del 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 Valkey o Redis OSS Lua. I tentativi di caricare uno script Lua più grande restituiscono un errore.                                                                                                                                                 |
| Dimensione dello slot           | 32 GiB       | La dimensione massima di un singolo hash slot Valkey o Redis. OSS I client che cercano di impostare più dati di questi su un singolo OSS slot Valkey o Redis attiveranno la politica di espulsione sullo slot e, se nessuna chiave è eliminabile, riceveranno un errore out of memory (). OOM |

## Cluster progettati autonomamente

Per i cluster progettati autonomamente, consulta [Parametri Valkey e Redis OSS](#) 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.

## IPv6 esempi di client per Valkey, Redis OSS e Memcached

ElastiCache è compatibile con Valkey, Redis OSS e Memcached. Ciò significa che i client che supportano IPv6 le connessioni dovrebbero essere in grado di connettersi a cluster IPv6 abilitati ElastiCache (Memcached). Ci sono alcune avvertenze che vale la pena notare quando si interagisce con risorse abilitate. IPv6

È possibile visualizzare il post di blog [sulle migliori pratiche per i clienti Valkey e Redis](#) sul blog AWS Database per consigli sulla configurazione dei client Valkey e Redis per le risorse. OSS ElastiCache

Di seguito sono riportate le migliori pratiche per interagire con le ElastiCache risorse IPv6 abilitate con le librerie client open source di uso comune.

### Client convalidati con Valkey e Redis OSS

ElastiCache è compatibile con Valkey e Redis open source. OSS Ciò significa che Valkey e OSS i client Redis open source che supportano IPv6 le connessioni dovrebbero essere in grado di connettersi a cluster IPv6 abilitati ElastiCache (Redis). OSS Inoltre, molti dei client Python e Java più diffusi sono stati specificamente testati e convalidati per funzionare con tutte le configurazioni di tipo di rete supportate (IPv4 solo, IPv6 solo e Dual Stack)

I seguenti client sono stati specificamente convalidati per funzionare con tutte le configurazioni dei tipi di rete supportate per Valkey e Redis. OSS

Clients convalidati:

- [Redis Py \(\) – 4,1.2](#)
- [Lettuce — Versione: 6.1.6. RELEASE](#)
- [Jedis – Versione: 3.6.0](#)

## Le migliori pratiche per i clienti (Valkey e RedisOSS)

Scopri le best practice per gli scenari più comuni e segui gli esempi di codice di alcune delle più popolari librerie OSS client open source di Valkey e Redis (redis-py e Lettuce)PHPRedis, nonché le migliori pratiche per interagire con le risorse con ElastiCache le librerie client Memcached open source di uso comune.

### Argomenti

- [Grande numero di connessioni \(Valkey e RedisOSS\)](#)
- [Individuazione dei client del cluster e backoff esponenziale \(Valkey e Redis\) OSS](#)
- [Configurare un timeout lato client \(Valkey e Redis\) OSS](#)
- [Configurare un timeout di inattività sul lato server \(Valkey e Redis\) OSS](#)
- [Script Lua](#)
- [Archiviazione di articoli composti di grandi dimensioni \(Valkey e RedisOSS\)](#)
- [Configurazione del client Lettuce \(Valkey e Redis\) OSS](#)
- [Configurazione di un protocollo preferito per i cluster dual stack \(Valkey e Redis\) OSS](#)

### Grande numero di connessioni (Valkey e RedisOSS)

Le cache serverless e i singoli nodi ElastiCache (RedisOSS) 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. Valkey e Redis dispongono OSS ciascuno di un processo a thread singolo basato su un ciclo di eventi in cui le richieste dei client in arrivo 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.

È possibile eseguire la seguente serie di azioni per evitare di imbattersi in un collo di bottiglia di connessione su un server Valkey o Redis: OSS

- Esegui le operazioni di lettura dalle repliche di lettura. Ciò può essere fatto utilizzando gli endpoint di ElastiCache lettura in modalità cluster disattivata o utilizzando repliche per le letture in modalità cluster abilitata, inclusa una cache serverless.
- Distribuisci il traffico di scrittura su più nodi primari. Sono disponibili due modi per farlo: È possibile utilizzare un cluster Valkey o Redis OSS con più sharded con un client compatibile con la modalità cluster. 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 TCP connessione è un'operazione computazionalmente costosa rispetto ai tipici comandi Valkey o Redis. OSS Ad esempio, la gestione di una GET richiestaSET/è un ordine di grandezza più veloce 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 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 (AmazonEC2) connessa a un nodo Graviton2 (m6g.2xlarge) (Redis). ElastiCache OSS 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 dei client del cluster e backoff esponenziale (Valkey e Redis) OSS

Quando ci si connette a un cluster ElastiCache Valkey o Redis in modalità OSS cluster abilitata, la libreria client corrispondente deve essere in grado di 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.
- Viene ricevuto un MOVED reindirizzamento dal server, ad esempio in caso di failover, quando tutti gli slot serviti dal precedente nodo primario vengono rilevati dalla replica, o in caso di ripartizionamento quando gli slot vengono spostati dal nodo primario di origine al nodo primario di destinazione

L'individuazione dei client viene in genere eseguita tramite l'invio di un CLUSTER NODE comando CLUSTER SLOT or al server Valkey o Redis. OSS Consigliamo CLUSTER SLOT questo metodo

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 CLUSTER SLOT comando 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 OSS server Valkey o Redis di rispondere per un periodo di tempo prolungato, con un utilizzo al 100%. CPU L'interruzione si prolunga se ogni CLUSTER SLOT comando 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 [Ricerca degli endpoint di connessione in ElastiCache](#) 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 esponenziale dei tentativi di backoff in redis-py e Lettuce. PHPRedis

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 tramite l'argomento fornito durante la retry\\_on\\_timeout creazione di un oggetto Redis. OSS](#) Di seguito è illustrato un meccanismo di ripetizione dei tentativi personalizzato con backoff esponenziale e jitter. Inviemo 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 di logica di backoff 2: PHPRedis

PHPRedis dispone di un meccanismo di riprova integrato che riprova 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. Abbiamo inviato una pull request per implementare nativamente il backoff esponenziale in PHPRedis\(#1986\), che da allora è stata unita e documentata.](#) Per chi utilizza l'ultima versione di PHPRedis, non sarà necessario implementarlo manualmente, ma abbiamo incluso qui il riferimento per quelli delle 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
 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();
}

}
}
```

## Configurare un timeout lato client (Valkey e Redis) OSS

### 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 OSS comandi Valkey o Redis possono essere più costosi dal punto di vista computazionale rispetto ad altri. Ad esempio, gli script Lua o le EXEC transazioniMULTI/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 EXEC transazioniMULTI/o script Lua costituiti da più comandi Valkey o Redis individuali OSS
- Lettura di valori di grandi dimensioni
- Esecuzione di operazioni di blocco come BLPOP

Nel caso di un'operazione di blocco comeBLPOP, la migliore pratica consiste nell'impostare il timeout del comando su un numero inferiore al timeout del socket.

Di seguito sono riportati esempi di codice per l'implementazione di un timeout lato client in redis-py e Lettuce. PHPRedis

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 di 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"); // BLP0P with 1 second timeout
} finally {
if (connection != null) {
connection.close();
}

if (client != null){
client.shutdown();
}
}
}
```

## Configurare un timeout di inattività sul lato server (Valkey e Redis) OSS

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 Valkey e Redis OSS](#). 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

Valkey e Redis OSS supportano più di 200 comandi, inclusi quelli per eseguire gli script Lua. Tuttavia, quando si tratta di script Lua, ci sono diverse insidie che possono influire sulla memoria e sulla disponibilità di Valkey o Redis. OSS

### Script Lua non parametrizzati

Ogni script Lua viene memorizzato nella cache sul server Valkey o Redis prima di essere eseguito. OSS Gli script Lua non parametrizzati sono unici, il che può portare il OSS server Valkey o Redis

a memorizzare un gran numero di script Lua e a consumare più memoria. Per mitigare questo problema, assicuratevi che tutti gli script Lua siano parametrizzati e che ripuliscano regolarmente gli script Lua memorizzati nella cache, se necessario. `SCRIPT FLUSH`

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 in modo atomico, quindi il completamento può richiedere più tempo rispetto a un normale comando Valkey o Redis. OSS 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 sta mutando può far sì che il server Valkey o OSS Redis non risponda per molto tempo. Per mitigare questo problema, evita gli script Lua a esecuzione prolungata e testali in un ambiente di preproduzione.

## Script Lua con scritture invisibili

Esistono alcuni modi in cui uno script Lua può continuare a scrivere nuovi dati in Valkey o Redis anche quando Valkey o Redis sono finiti: OSS `OSS maxmemory`

- Lo script si avvia quando il OSS server Valkey o Redis si trova al di sotto `maxmemory` e contiene più operazioni di scrittura al suo interno
- Il primo comando di scrittura dello script non consuma memoria (ad esempio `DEL`), seguito da altre operazioni di scrittura che consumano memoria

- È possibile mitigare questo problema configurando una politica di sfratto adeguata in un server Valkey o Redis diverso da. OSS noeviction Ciò consente a Redis di eliminare gli oggetti e OSS liberare memoria tra gli script Lua.

## Archiviazione di articoli composti di grandi dimensioni (Valkey e RedisOSS)

In alcuni scenari, un'applicazione può archiviare elementi composti di grandi dimensioni in Valkey o Redis OSS (come un set di dati hash da più GB). Questa non è una pratica consigliata perché spesso porta a problemi di prestazioni in Valkey o Redis. OSS Ad esempio, il client può eseguire un HGETALL comando per recuperare l'intera raccolta di hash da più GB. Ciò può generare una notevole pressione sulla memoria del OSS server Valkey o Redis che memorizza nel buffer di output dell'elemento di grandi dimensioni nel buffer di output del client. Inoltre, per la migrazione degli slot in modalità cluster, ElastiCache non esegue la migrazione degli 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 MGET comando per recuperare più valori-chiave 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 dei HMGET comandi HGET or anziché su raccolte hash. HGETALL

## Configurazione del client Lettuce (Valkey e Redis) OSS

Questa sezione descrive le opzioni di configurazione consigliate per 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, abilitata TLS](#)
- [Esempio: la configurazione Lettuce per la modalità cluster è disattivata, abilitata TLS](#)

## Cache Java DNS TTL

La macchina virtuale Java (JVM) memorizza nella cache le ricerche dei DNS nomi. Quando JVM risolve un nome host in un indirizzo IP, memorizza l'indirizzo IP nella cache per un periodo di tempo specificato, noto come `(.)`. `time-to-liveTTL`

La scelta del TTL valore è un compromesso tra latenza e reattività al cambiamento. Con tempi più breviTTLs, i DNS resolver notano gli aggiornamenti nel cluster più velocemente. DNS 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 TTL è troppo basso, aumenta il volume delle query, il che può aumentare la latenza dell'applicazione. Sebbene non esista un TTL valore corretto, vale la pena considerare il periodo di tempo che ci si può permettere di attendere che una modifica abbia effetto quando si imposta il TTL valore.

Poiché ElastiCache i nodi utilizzano voci di DNS nome che potrebbero cambiare, ti consigliamo TTL di configurarle JVM con un minimo da 5 a 10 secondi. In questo modo, quando l'indirizzo IP di un nodo cambia, l'applicazione sarà in grado di ricevere e utilizzare il nuovo indirizzo IP della risorsa richiedendo l'DNSimmissione.

In alcune configurazioni Java, l'JVMimpostazione predefinita TTL è impostata in modo che non aggiorni mai le DNS voci fino al riavvio di. JVM

Per dettagli su come impostare il tuo JVMTTL, vedi [Come](#) impostare il. JVM TTL

## 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 DNS relativa ricerca URI restituisce un elenco di tutti i nodi disponibili nel cluster, e viene risolta casualmente su uno di essi durante l'inizializzazione del cluster. Per ulteriori dettagli su come funziona l'aggiornamento della topologia, vedere più avanti in questo argomento. `dynamicRefreshResources`

## SocketOption

Abilita. [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 per la modalità cluster

Abilita in [AutoReconnect](#) caso di interruzione della connessione.

Impostare [CommandTimeout](#). Per ulteriori dettagli, consulta la sezione relativa ai timeout più avanti in questo argomento.

Impostato [nodeFilter](#) per filtrare i nodi guasti dalla topologia. Lettuce salva tutti i nodi che si trovano nell'output dei «nodi del cluster» (inclusi i nodi con PFAIL/FAILstatus) nelle «partizioni» del client (note anche come shard). 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, dopo il completamento di un failover e l'avvio del processo di ripristino da parte del cluster, mentre clusterTopology viene aggiornato, la mappa dei nodi del bus del cluster trascorre un breve periodo di tempo in cui il nodo inattivo viene elencato come nodo, prima di essere completamente rimosso FAIL dalla topologia. Durante questo periodo, il client Lettuce 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` set to `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 iniziali (quando `DynamicRefreshSources` è vero) riduce la probabilità che si verifichi questo problema, poiché almeno alcuni nodi iniziali dovrebbero avere una vista topologica aggiornata dopo un failover con il primario appena promosso.

`ClusterTopologyRefreshOptions`: opzioni per controllare l'aggiornamento della topologia del cluster del client Cluster Mode Enabled

**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 Lettuce's. `MasterSlaveTopologyRefresh` 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 Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#).

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.

Abilita. [enablePeriodicRefresh](#) Ciò consente gli aggiornamenti periodici della topologia del cluster in modo che il client aggiorni la topologia del cluster a intervalli di `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.

[enableAllAdaptiveRefreshTriggers](#) Abilita. Ciò consente l'aggiornamento adattivo della topologia che utilizza tutti i [trigger](#): `MOVED _REDIRECT, _, ASK _REDIRECTRECONNECTS, PERSISTENT UNCOVERED _`. `SLOT UNKNOWN NODE` I trigger di aggiornamento adattivi avviano gli aggiornamenti della visualizzazione della topologia in base agli eventi che si verificano durante le operazioni dei cluster Valkey o Redis. OSS 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).

Abilita. [closeStaleConnections](#) Ciò consente di chiudere le connessioni obsolete durante l'aggiornamento della topologia del cluster. Entra in vigore solo se [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled \(\)](#) è vero. 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.

Abilita [dynamicRefreshResources](#). Ti consigliamo di abilitarlo `dynamicRefreshResources` per i cluster di piccole dimensioni e di disabilitarlo per i cluster di grandi dimensioni. `dynamicRefreshResources` consente di scoprire i nodi del cluster dal nodo iniziale 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 vero, Lettuce

interroga tutti i nodi per ottenere la visualizzazione 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

Configura con [DnsResolverDirContextDnsResolver](#) Il DNS resolver è basato su com.sun.jndi.dns di Java. DnsContextFactory.

Configurazione [reconnectDelay](#) con backoff esponenziale e full jitter. Lettuce ha meccanismi di ripetizione incorporati basati sulle strategie di backoff esponenziale. Per maggiori dettagli, consulta [Exponential Backoff](#) and Jitter sul blog di architettura. AWS Per ulteriori informazioni sull'importanza di adottare una strategia di backoff tra tentativi ripetuti, consulta le sezioni sulla logica di backoff del post sulle [best practice](#) sul Database Blog. 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, utilizzate un timeout più lungo

per i comandi che eseguono iterazioni su più tasti,, o gli FLUSHDB script FLUSHALL Lua. KEYS SMEMBERS Utilizzate timeout più brevi per i comandi a chiave singola, come, e. SET GET HSET

### Note

I timeout configurati nell'esempio seguente sono per i test che hanno eseguito GET comandi SET/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, abilitata TLS

### Note

I timeout nell'esempio seguente si riferiscono ai test che hanno eseguito GET comandi SET/ 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 setJVMPProperties() {
 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: la configurazione Lettuce per la modalità cluster è disattivata, abilitata TLS

### Note

I timeout nell'esempio seguente si riferiscono ai test che hanno eseguito GET comandiSET/ 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 setJVMPProperties() {
 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);
```

## Configurazione di un protocollo preferito per i cluster dual stack (Valkey e Redis) OSS

Per i cluster Valkey o Redis abilitati alla modalità OSS cluster, è possibile controllare il protocollo che i client utilizzeranno per connettersi ai nodi del cluster con il parametro IP Discovery. Il parametro IP Discovery può essere impostato su uno o. IPv4 IPv6

Per OSS i cluster Valkey o Redis, il parametro IP Discovery imposta il protocollo IP utilizzato nell'output degli [slot del cluster \(\)](#), dei [cluster shard \(\)](#) e [dei nodi del cluster \(\)](#). Questi comandi vengono utilizzati dai client per scoprire la topologia del cluster. I client utilizzano i comandi IPs in theses per connettersi agli altri nodi del 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 le modifiche si sono completamente propagate per un OSS cluster Valkey o Redis, monitora l'output di. `cluster slots` Una volta che tutti i nodi restituiti dal comando `cluster slots` hanno riportato il nuovo protocollo, IPs le modifiche hanno terminato la propagazione.

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 targetType = InetAddress.class; // Or InetAddress.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 finalTargetType.isInstance(InetAddress.getByName(node));
 } catch (UnknownHostException ignored) {}
 return false;
}));
```

## Le migliori pratiche per i clienti (Memcached)

### Configurazione ElastiCache del client per un bilanciamento efficiente del carico (Memcached)

#### Note

Questa sezione si applica ai cluster Memcached a più nodi progettati autonomamente.

Per utilizzare in modo efficace più nodi ElastiCache Memcached, devi essere in grado di distribuire le chiavi della cache tra i nodi. Un modo semplice per eseguire il bilanciamento del carico di un cluster con  $n$  nodi consiste nel calcolare l'hash della chiave dell'oggetto ed eseguire il comando `mod` sul risultato per  $n - \text{hash}(\text{key}) \bmod n$ . Il valore risultante (da 0 a  $n-1$ ) è il numero del nodo in cui inserisci l'oggetto.

Questo approccio è semplice ed efficace, purché il numero dei nodi ( $n$ ) sia costante. Quando aggiungi o rimuovi un nodo dal cluster, il numero di chiavi da spostare corrisponde tuttavia a  $(n -$

1) / n (dove n è il nuovo numero di nodi). Con questo approccio, il numero di chiavi da spostare è pertanto maggiore, con un conseguente numero elevato di mancati riscontri nella cache iniziali, specialmente man mano che aumenta il numero dei nodi. Con il dimensionamento da 1 a 2 nodi, viene spostato un numero di chiavi pari a  $(2-1)/2$  (50 percento), ovvero il caso migliore. Con il dimensionamento da 9 a 10 nodi, viene spostato un numero di chiavi pari a  $(10-1)/10$  (90 percento). In caso di un dimensionamento dovuto a un picco nel traffico, è bene evitare un elevato numero di mancati riscontri nella cache, in quanto ciò causerebbe hit nel database, già sovraccaricato dai picchi di traffico.

La soluzione a tale problema è rappresentata da un hashing coerente. L'hashing coerente utilizza un algoritmo in modo tale che quando un nodo viene aggiunto o rimosso da un cluster, il numero di chiavi da spostare è circa  $1/n$  (dove n è il nuovo numero di nodi). Passando da 1 a 2 nodi, viene spostato un numero di chiavi pari a  $1/2$  (50 percento), ovvero il caso peggiore. Passando da 9 a 10 nodi, viene spostato un numero di chiavi pari a  $1/10$  (10 percento).

Sei tu, in qualità di utente, che controlli l'algoritmo di hashing utilizzato per il cluster a più nodi. Ti consigliamo di configurare i client in modo da utilizzare un hashing coerente. Fortunatamente, sono disponibili numerose librerie client Memcached nei linguaggi più comuni in grado di implementare l'hashing coerente. Consulta la documentazione della libreria che utilizzi per verificare se supporta l'hashing coerente e ottenere informazioni su come implementarlo.

Se lavori in JavaPHP, o. NET, ti consigliamo di utilizzare una delle librerie ElastiCache client Amazon.

### Hashing coerente in Java

Il client Java ElastiCache Memcached si basa sul client Java open source spymemcached, che dispone di funzionalità di hashing coerenti integrate. La libreria include una classe che implementa un hashing coerente. `KetamaConnectionFactory` Per impostazione predefinita, l'hashing coerente è disattivato in spymemcached.

Per ulteriori informazioni, consultate la `KetamaConnectionFactory` documentazione all'indirizzo.

[KetamaConnectionFactory](#)

### Hashing coerente PHP con Memcached

Il PHP client ElastiCache Memcached è un wrapper della libreria Memcached integrata. PHP Per impostazione predefinita, l'hashing coerente è disattivato dalla libreria Memcached. PHP

Per attivare l'hashing coerente, utilizza il seguente codice.

```
$m = new Memcached();
```

```
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

Oltre al codice precedente, ti consigliamo anche di attivare `memcached.sess_consistent_hash` nel file `php.ini`.

[Per ulteriori informazioni, consulta la documentazione sulla configurazione in fase di esecuzione per Memcached all'indirizzo http://php.net/manual/en/memcached.configuration.php](http://php.net/manual/en/memcached.configuration.php). PHP Nota in particolare il parametro `memcached.sess_consistent_hash`.

### Utilizzo di hashing coerente. NETcon Memcached

Il ElastiCache Memcached. NETil client è un involucro di Enyim Memcached. Per impostazione predefinita, l'hashing coerente è attivato dal client Enyim Memcached.

[Per maggiori informazioni, consulta la documentazione in /wiki/ -Configuration#. memcached/locator https://github.com/enyim/EnyimMemcached MemcachedClient user-content-memcachedlocator](https://github.com/enyim/EnyimMemcached)

## Client convalidati con Memcached

I seguenti client sono stati specificamente convalidati per funzionare con tutte le configurazioni dei tipi di rete supportate per Memcached.

Clients convalidati:

- [AWS ElastiCache Cluster Client Memcached per Php — versione \\*3.6.2](#)
- [AWS ElastiCache Cluster Client Memcached](#) per Java: master più recente su Github

## Configurazione di un protocollo preferito per i cluster dual stack (Memcached)

Per cluster Memcached 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 o. IPv4 IPv6

Il parametro IP Discovery controlla il protocollo IP utilizzato nell'output del cluster config get. Il che a sua volta determinerà il protocollo IP utilizzato dai client che supportano l'individuazione automatica per i cluster ElastiCache (Memcached).

La modifica di IP Discovery non comporterà alcun tempo di inattività per i client connessi. Tuttavia, la propagazione delle modifiche richiederà tempo.

Monitorare l'output di `getAvailableNodeEndpoints` per Java e per Php monitorare l'output di `getServerList`. Una volta che l'output di queste funzioni viene segnalato IPs per tutti i nodi del cluster che utilizzano il protocollo aggiornato, le modifiche hanno terminato la propagazione.

### Esempio di Java

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

do {
 nodes =
 client.getAvailableNodeEndpoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors.toSet());

 Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
 try {
 return finalTargetProtocolType.isInstance(InetAddress.getByAddress(node));
 } catch (UnknownHostException ignored) {}
 return false;
})));
```

### Esempio di Php:

```
$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
 # The PHP memcached client only updates the server list if the polling interval has
 expired and a
 # command is sent
 $client->get('test');

 $nodes = $client->getServerList();

 sleep(1);
 $target_ips_count = 0;
```

```
// For IPv4 use FILTER_FLAG_IPV4
$target_ips_count = count(array_filter($nodes, function($node) { return
filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));

} while (count($nodes) !== $target_ips_count);
```

Le eventuali connessioni client esistenti create prima dell'aggiornamento di IP Discovery, verranno comunque connesse utilizzando il vecchio protocollo. Tutti i client convalidati si riconnetteranno automaticamente al cluster utilizzando il nuovo protocollo IP una volta rilevate le modifiche nell'output dei comandi di individuazione del cluster. Tuttavia, ciò dipende dall'implementazione del client.

## TLScluster dual stack ElastiCache abilitati

Quando TLS è abilitato per ElastiCache i cluster `cluster slots`, `cluster shards` le funzioni di rilevamento del cluster (e `cluster nodes` per Redis) o `config get cluster` per Memcached restituiscono nomi di host anziché IP. I nomi host vengono quindi utilizzati anziché IP per connettersi al cluster ed eseguire una stretta di mano. ElastiCache TLS Ciò significa che i client non sono interessati dal parametro Individuazione IP. Per i cluster TLS abilitati, il parametro IP Discovery non ha alcun effetto sul protocollo IP preferito. Invece, il protocollo IP utilizzato sarà determinato dal protocollo IP preferito dal client per la risoluzione dei nomi host. DNS

### Client Java

Quando ci si connette da un ambiente Java che supporta entrambi IPv4 e IPv6, IPv6 per impostazione predefinita Java preferisce la compatibilità con le IPv4 versioni precedenti. Tuttavia, la preferenza del protocollo IP è configurabile tramite gli JVM argomenti. Preferire IPv4, JVM accettare `-Djava.net.preferIPv4Stack=true` e preferire IPv6 impostati `-Djava.net.preferIPv6Stack=true`. L'impostazione `-Djava.net.preferIPv4Stack=true` significa che non JVM effettueranno più alcuna IPv6 connessione. Per Valkey o RedisOSS, ciò include quelle verso altre applicazioni non Valkey e non Redis. OSS

### Preferenze a livello di host

In generale, se il client o il runtime del client non forniscono opzioni di configurazione per impostare una preferenza del protocollo IP, quando si esegue la DNS risoluzione, il protocollo IP dipenderà dalla configurazione dell'host. Per impostazione predefinita, la maggior parte degli host IPv6 preferisce questa preferenza, IPv4 ma questa preferenza può essere configurata a livello di host. Ciò influirà su tutte DNS le richieste provenienti dall'host, non solo su quelle ai ElastiCache cluster.

## 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 che preferisci IPv4 quando ti connetti a un ElastiCache cluster, aggiorna la precedenza per l'CIDRintervallo che comprende il cluster in modo che sia superiore IPs alla precedenza per le connessioni predefinite. IPv6 Per impostazione predefinita, IPv6 le connessioni hanno una precedenza di 40. Ad esempio, supponendo che il cluster si trovi in una sottorete con CIDR `172.31.0.0/16`, la configurazione seguente farebbe sì che i client preferiscano le connessioni a quel cluster. IPv4

```
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 politiche di preferenza in modo da preferire le connessioni alle IPv4 connessioni per l'intervallo specificato. IPv6 CIDR Ad esempio, supponendo che il cluster si trovi in una sottorete con CIDR esecuzione `172.31.0.0:0/16`, si otterrà la seguente tabella di precedenza che `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` farebbe sì che i client preferiscano la connessione al cluster. IPv4

```
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
```

## Gestione della memoria riservata per Valkey e Redis OSS

La memoria prenotata è la memoria destinata all'utilizzo diverso dai dati. Quando eseguono un backup o un failover, Valkey e Redis OSS utilizzano la memoria disponibile per registrare le operazioni di scrittura sul 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, è possibile trovare informazioni sulle opzioni per la gestione della memoria riservata per ElastiCache (RedisOSS) e su come applicarle.

### 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 utilizzi una versione di Redis OSS precedente alla 2.8.22, riserva più memoria per i backup e i failover rispetto a Redis 2.8.22 o successiva. OSS Questo requisito è dovuto ai diversi modi in cui (Redis) implementa il processo di backup. ElastiCache OSS La regola generale è quella di riservare metà del maxmemory valore di un tipo di nodo per Redis OSS overhead per le versioni precedenti alla 2.8.22 e un quarto per le versioni Redis 2.8.22 e successive. OSS

A causa delle diverse modalità di ElastiCache implementazione del processo di backup e replica, la regola generale consiste nel riservare il 25% del valore di un tipo di nodo utilizzando il parametro. maxmemory reserved-memory-percent Questo è il valore predefinito e consigliato nella maggior parte dei casi.

Quando i tipi di istanze micro e piccole istanze burstable funzionano vicino maxmemory ai limiti, è possibile che si verifichi l'utilizzo dello swap. Per migliorare l'affidabilità operativa di questi tipi di istanze durante il backup, la replica e il traffico elevato, consigliamo di aumentare il valore del reserved-memory-percent parametro fino al 30% sui tipi di istanze di piccole dimensioni e fino al 50% sui tipi di micro istanze.

Per carichi di lavoro impegnativi in scrittura su ElastiCache cluster con suddivisione dei dati su più livelli, consigliamo di aumentare la memoria disponibile del nodo fino reserved-memory-percent al 50%.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS](#)
- [Modalità di implementazione di sincronizzazione e backup](#)
- [Suddivisione dei dati su più livelli in ElastiCache](#)

## Parametri per gestire memoria prenotata

A partire dal 16 marzo 2017, Amazon ElastiCache fornisce due parametri che si escludono a vicenda per la gestione della memoria Valkey o Redis OSS e. `reserved-memory` `reserved-memory-percent` Nessuno di questi parametri fa parte della distribuzione Valkey o Redis. OSS

A seconda di quando sei diventato ElastiCache cliente, l'uno o l'altro di questi parametri è il parametro di gestione della memoria predefinito. Questo parametro si applica quando si crea un nuovo OSS cluster o gruppo di replica Valkey o Redis e si utilizza un gruppo di parametri predefinito.

- Per i clienti che hanno iniziato prima del 16 marzo 2017: quando si crea un OSS cluster o un gruppo di replica Redis utilizzando il gruppo di parametri predefinito, 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 dopo tale data: quando si crea un OSS cluster o un gruppo di replica Valkey o Redis utilizzando il gruppo di parametri predefinito, 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 OSS memoria Valkey o Redis, potresti preferire utilizzare quello che non è quello predefinito 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. È quindi possibile utilizzare il gruppo di parametri personalizzato ogni volta che si crea un nuovo cluster o gruppo di replica Valkey o RedisOSS. 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 ElastiCache parametri](#)
- [Modifica di un gruppo di ElastiCache 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 riservata ElastiCache (RedisOSS) veniva eseguita utilizzando il parametro `reserved-memory`. Il valore di default di `reserved-memory` è 0. Questa impostazione predefinita non riserva memoria per il OSS sovraccarico di Valkey o Redis e consente a Valkey o Redis OSS di consumare 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 personalizzati, viene impostato `reserved-memory` un valore appropriato per la OSS versione Valkey o 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` è specifico ElastiCache e non fa parte della distribuzione generale di RedisOSS.

La procedura seguente mostra come utilizzare per `reserved-memory` gestire la memoria sul cluster Valkey o RedisOSS.

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 ElastiCache parametri](#).

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name redis6x-m3x1 \
 --description "Redis OSS 2.8.x for m3.xlarge node type" \
 --cache-parameter-group-family redis6.x
```

2. Calcola quanti byte di memoria riservare per il sovraccarico di Valkey o Redis. OSS Il valore di `maxmemory` per il tipo di nodo è disponibile in [Parametri specifici del tipo di nodo Redis OSS](#).
3. Modificare il gruppo di parametri personalizzato in modo che il parametro `reserved-memory` corrisponda al numero di byte calcolati nella fase precedente. L' AWS CLI esempio seguente presuppone che stiate utilizzando una versione di Redis OSS precedente alla 2.8.22 e che

dobbiare riservare metà di quella del nodo. `maxmemory` Per ulteriori informazioni, consulta [Modifica di un gruppo di ElastiCache 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. Modifica il OSS cluster o il gruppo di replica Redis per utilizzare il gruppo di parametri personalizzato.

L'CLlesempio seguente modifica il cluster `my-redis-cluster` per utilizzare il gruppo di parametri personalizzato a partire da subito. `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'CLlesempio seguente modifica il gruppo di replica `my-redis-repl-grp` per utilizzare il gruppo di parametri personalizzati a partire da subito. `redis28-m3x1` 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-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 (RedisOSS). 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` è specifico ElastiCache e non fa parte della distribuzione generale di RedisOSS.

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 [Suddivisione dei dati su più livelli in ElastiCache](#).

Per riservare memoria utilizzando `reserved-memory-percent`

Da utilizzare `reserved-memory-percent` per gestire la memoria del cluster ElastiCache (RedisOSS), esegui una delle seguenti operazioni:

- Se utilizzi Redis OSS 2.8.22 o versione successiva, assegna il gruppo di parametri predefinito al cluster. Il valore di default di 25 per cento dovrebbe essere adeguato. In caso contrario, attenersi alla procedura descritta di seguito per modificare il valore.
- Se utilizzi una versione di Redis OSS precedente alla 2.8.22, probabilmente dovrai riservare più memoria del 25% predefinito. `reserved-memory-percent` A tale scopo, procedi come indicato di seguito.

Per modificare il valore percentuale di `reserved-memory-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 ElastiCache parametri](#).

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name redis28-50 \
 --description "Redis OSS 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 ElastiCache parametri](#).

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis28-50 \
 --parameter-name-values "ParameterName=reserved-memory-percent,
 ParameterValue=50"
```

3. Utilizza questo gruppo di parametri personalizzato per qualsiasi OSS cluster o gruppo di replica Redis che esegue una versione di Redis OSS precedente alla 2.8.22.

L'CLlesempio seguente modifica il OSS cluster Redis per utilizzare il gruppo di parametri personalizzato `my-redis-cluster` a partire da subito. `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'CLlesempio seguente modifica il gruppo di OSS replica Redis `my-redis-repl-grp` per utilizzare il gruppo di parametri personalizzato a partire da subito. `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 eri un ElastiCache cliente attuale il 16 marzo 2017, il tuo parametro di gestione della memoria riservata predefinito è `reserved-memory` pari a zero (0) byte di memoria riservata. Se sei diventato ElastiCache cliente dopo il 16 marzo 2017, il tuo parametro di gestione della memoria riservata predefinito prevede `reserved-memory-percent` il 25 percento della memoria riservata del nodo. Questo vale indipendentemente dal momento in cui hai creato il cluster ElastiCache (RedisOSS) o

il gruppo di replica. Tuttavia, è possibile modificare il parametro di gestione della memoria riservata utilizzando o. AWS CLI ElastiCache API

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 ElastiCache parametri](#).

Per specificare `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'CLlesempio seguente modifica il gruppo di parametri personalizzati `redis32-cluster-on` in modo da utilizzarlo `reserved-memory-percent` per gestire la memoria riservata. 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 ElastiCache 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'CLlesempio seguente modifica il gruppo di parametri personalizzati `redis32-m3x1` in modo che venga utilizzato `reserved-memory` per gestire la memoria riservata. 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 ElastiCache parametri](#).



```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis32-m3x1 \
 --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

## Le migliori pratiche per lavorare con cluster progettati autonomamente da Valkey e Redis OSS

L'uso della tecnologia Multi-AZ, la disponibilità di memoria sufficiente, il ridimensionamento dei cluster e la riduzione al minimo dei tempi di inattività sono tutti concetti utili da tenere a mente quando si lavora con cluster progettati autonomamente in Valkey o Redis. OSS Ti consigliamo di esaminare e seguire queste best practices.

### Argomenti

- [Ridurre al minimo i tempi di inattività con la funzione Multi-AZ](#)
- [Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS](#)
- [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

Esistono diversi casi in cui ElastiCache Valkey o Redis OSS potrebbero dover sostituire un nodo primario; questi includono alcuni tipi di manutenzione pianificata e l'improbabile evento di un guasto del nodo primario o della 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 ed effettuare il provisioning di un nuovo nodo primario, in quanto ElastiCache gestirà il 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.

Vedi [Riduzione al minimo dei tempi di inattività ElastiCache utilizzando Multi-AZ con Valkey e Redis OSS](#), per saperne di più su Multi-AZ e sulla riduzione al minimo dei tempi di inattività.

## Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS

Istantanee e sincronizzazioni in Valkey 7.2 e versioni successive e Redis versione 2.8.22 e successive OSS

Valkey offre il supporto predefinito per istantanee e sincronizzazioni. Redis OSS 2.8.22 introduce un processo di salvataggio senza forkless che consente di allocare più memoria all'uso dell'applicazione senza incorrere in un maggiore utilizzo dello swap durante le sincronizzazioni e i salvataggi. Per ulteriori informazioni, consulta [Modalità di implementazione di sincronizzazione e backup](#).

Istantanee e sincronizzazioni Redis precedenti alla versione OSS 2.8.22

Quando lavori con ElastiCache (RedisOSS), Redis OSS chiama un comando di scrittura in background in diversi casi:

- Durante la creazione di uno snapshot per un backup.
- Durante la sincronizzazione delle repliche con il nodo principale in un gruppo di repliche
- Quando si abilita la funzione append-only file (AOF) per Redis. OSS
- Durante la promozione di una replica a primaria (che causa una sincronizzazione tra nodi principali/replica).

Ogni volta che Redis OSS esegue un processo di scrittura in background, è necessario disporre di memoria disponibile sufficiente per far fronte al sovraccarico del processo. La mancanza di memoria sufficiente provoca un errore del processo. Per questo motivo, è importante scegliere un tipo di istanza di nodo con memoria sufficiente durante la creazione del cluster Redis. OSS

Processo di scrittura in background e utilizzo della memoria con Valkey e Redis OSS

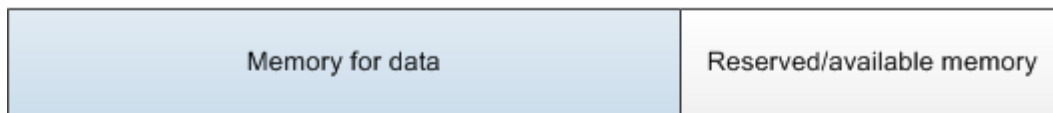
Ogni volta che viene chiamato un processo di scrittura in background, Valkey e Redis ne eseguono un OSS fork (ricordate che questi motori sono a thread singolo). Un fork mantiene i dati su disco in un file snapshot Redis .rdb. OSS Tutti gli altri servizi fork eseguono operazioni di lettura e scrittura. Per garantire che l'istanza sia un'istanza point-in-time, tutti gli aggiornamenti e le aggiunte ai 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 verifici uno dei seguenti casi:

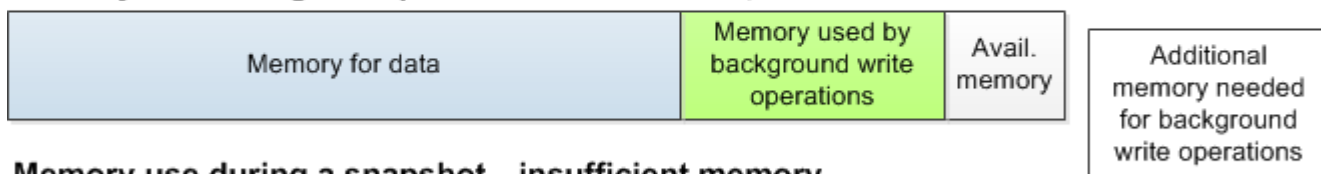
- L'applicazione esegue molte operazioni di scrittura e richiede pertanto un'elevata quantità di memoria disponibile per accettare dati nuovi o aggiornati.
- Hai a disposizione una quantità di memoria veramente limitata in cui scrivere dati nuovi o aggiornati.
- Il set di dati di cui disponi è di grandi 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.

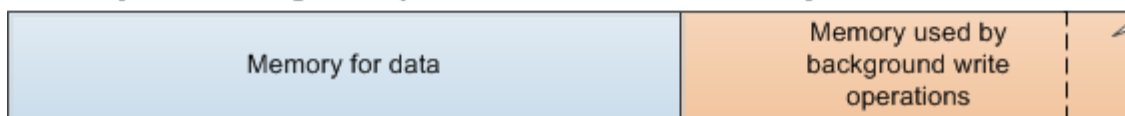
### Memory use prior to a snapshot



### 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 su come Valkey e Redis OSS eseguono le istantanee, consulta <http://valkey.io>.

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

Come evitare di esaurire la memoria durante l'esecuzione di una scrittura in background

Ogni volta che BGREWRITEAOF viene chiamato un processo di scrittura in background come BGSAVE o, per evitare che il processo abbia esito negativo, è necessario disporre di più memoria di quella che verrà consumata dalle operazioni di scrittura durante il processo. Lo scenario peggiore è che durante l'operazione di scrittura in background ogni record venga aggiornato e alcuni nuovi record vengano aggiunti alla cache. Per questo motivo, si consiglia di impostare su 50 (50 per cento) per

OSS le versioni Redis precedenti `reserved-memory-percent` alla 2.8.22 o su 25 (25 percento) per Valkey e tutte le versioni Redis 2.8.22 e successive. OSS

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 predefinito per `reserved-memory` è 0, che consente OSS a Redis di consumare tutta la memoria massima con i dati, lasciando potenzialmente troppo poca memoria per altri usi, come un processo di scrittura in background. Per i valori di `maxmemory` per tipo di istanza del nodo, consulta [Parametri specifici del tipo di nodo Redis OSS](#).

È inoltre possibile utilizzare il `reserved-memory` parametro per ridurre la quantità di memoria utilizzata sulla confezione.

Per ulteriori informazioni sui parametri specifici di Valkey e Redis in, vedere. ElastiCache [Parametri Valkey e Redis OSS](#)

Per informazioni sulla creazione e sulla modifica di gruppi di parametri, consulta [Creazione di un gruppo di ElastiCache parametri](#) e [Modifica di un gruppo di ElastiCache parametri](#).

## 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 attenerci a tutte le best practice relative al cluster per una distribuzione uniforme dei modelli di carico di lavoro. È inoltre consigliabile completare i passaggi indicati di seguito.

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

- Testa la tua applicazione - Testa il comportamento della tua applicazione durante il `resharding` in un ambiente di gestione temporanea, se possibile.
- Ottieni una notifica immediata dei problemi di dimensionamento - Il `resharding` è un'operazione che richiede notevoli risorse di calcolo. Per questo motivo, consigliamo di mantenere CPU l'utilizzo al di sotto dell'80% sulle istanze multicore e meno del 50% sulle istanze single core durante il `resharding`. Monitora le metriche ElastiCache (RedisOSS) e avvia il `resharding` prima che l'applicazione inizi a rilevare problemi di scalabilità. 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 shard: durante la scalabilità orizzontale del cluster, ElastiCache replicherà automaticamente le funzioni caricate in uno dei nodi esistenti (selezionati a caso) sui nuovi nodi. Se il tuo cluster ha Valkey 7.2 e versioni successive o Redis OSS 7.0 o versioni successive e la tua applicazione utilizza [Functions](#), ti consigliamo di caricare tutte le tue funzioni su tutti gli shard prima di eseguire la scalabilità orizzontale in modo che il cluster non finisca con funzioni diverse su shard diversi.

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 OSS 6, il BRPOPLPUSH comando non era supportato se opera 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 AutoFailover 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. Se AutoFailover disabilitato, il ruolo del nodo non cambia. Tuttavia, il tempo di inattività nelle operazioni gestite o non gestite è maggiore rispetto ai cluster abilitati. AutoFailover

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.

# Strategie di caching per Memcached

Nel seguente argomento, puoi trovare le strategie per popolare e gestire la cache Memcached.

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](#)
- [Aggiungendo 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 trova tra l'applicazione e l'archivio dati (database) a cui accede. Ogni volta che l'applicazione richiede dati, invia innanzitutto la richiesta alla cache. ElastiCache Se i dati esistono nella cache e sono aggiornati, ElastiCache restituisce i 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 [Aggiungendo 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. [Aggiungendo un valore time to live \(TTL\)](#), puoi ridurre al minimo lo spreco 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"})
```

## Aggiungendo 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 time to live (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.

Time to live (TTL) è un valore intero che specifica il numero di secondi che mancano alla scadenza della chiave. Valkey o Redis OSS possono specificare secondi o millisecondi per questo valore. Memcached specifica questo valore in secondi. 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 i comandi Valkey e Redis OSS o i comandi Memcached. set](#)

### TTL esempi di pseudocodice

Quello che segue è un esempio di pseudocodice di logica di scrittura 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
```

Di seguito è riportato un esempio di pseudocodice di logica di lazy loading 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](#)
- [Ridimensionamento ElastiCache](#)

## Gestione del cluster progettato autonomamente in ElastiCache

ElastiCache offre due opzioni di implementazione, caching senza server e cluster progettati autonomamente. Ciascuno ha le proprie capacità e requisiti.

Questa sezione contiene argomenti per aiutarti a gestire i cluster progettati autonomamente.

### Note

Questi argomenti non si applicano a Serverless. ElastiCache

## Argomenti

- [Auto Scaling dei cluster Valkey e Redis OSS](#)
- [Modifica della modalità cluster](#)
- [Replica tra AWS regioni utilizzando datastore globali](#)
- [Alta disponibilità utilizzando gruppi di replica](#)
- [Gestione della manutenzione dei ElastiCache cluster](#)
- [Configurazione dei parametri del motore utilizzando gruppi di ElastiCache parametri](#)

## Auto Scaling dei cluster Valkey e Redis OSS

### Prerequisiti

ElastiCache L'Auto Scaling è limitato a quanto segue:

- Cluster Valkey o Redis OSS (modalità cluster abilitata) che eseguono Valkey 7.2 e versioni successive o che eseguono il motore Redis dalla versione 6.0 in poi OSS
- Cluster di data tiering (modalità cluster abilitata) che eseguono Valkey 7.2 e versioni successive o che eseguono il motore Redis dalla versione 7.0.7 in poi OSS
- Dimensioni XLarge delle istanze: grandi, 2 XLarge
- Famiglie di tipi di istanza: R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- L'Auto Scaling in non ElastiCache è supportato per i cluster in esecuzione in datastore globali, Outposts o Local Zones.

## Gestione automatica della capacità con ElastiCache Auto Scaling con Valkey o Redis OSS

ElastiCache la scalabilità automatica con Valkey o Redis OSS è la capacità di aumentare o diminuire automaticamente gli shard o le repliche desiderati nel servizio. ElastiCache ElastiCache 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 la scalabilità automatica utilizza la policy per aumentare o diminuire il numero di istanze in risposta ai carichi di lavoro effettivi.

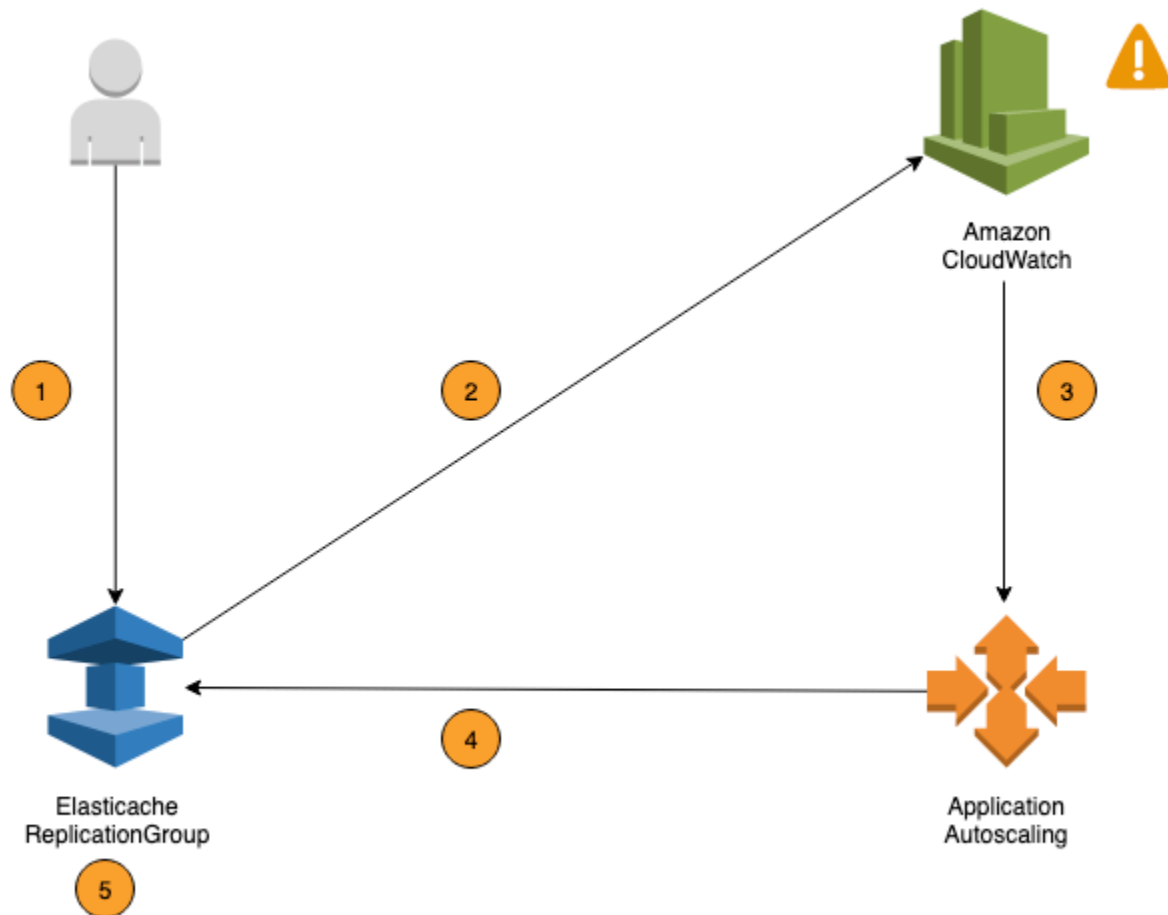
È possibile utilizzare il AWS Management Console per applicare una politica di scalabilità basata su una metrica predefinita. Un `predefined metric` viene definito in un'enumerazione in modo che tu possa specificarlo per nome nel codice o utilizzarlo nella AWS Management Console. I parametri personalizzati non possono essere selezionati tramite AWS Management Console. In alternativa, puoi utilizzare AWS CLI o Application Auto Scaling API per applicare una politica di scalabilità basata su una metrica predefinita o personalizzata.

ElastiCache con Valkey o Redis OSS supporta la scalabilità per le seguenti dimensioni:

- **Partizioni** : Aggiungi/rimuovi automaticamente frammenti nel cluster in modo simile al risharding online manuale. In questo caso, il ridimensionamento ElastiCache automatico attiva la scalabilità per tuo conto.
- **Repliche** : Aggiungi/rimuovi automaticamente le repliche nel cluster in modo simile alle operazioni manuali Aumenta/Diminuisci. ElastiCache con la scalabilità OSS automatica Valkey o Redis aggiunge/rimuove le repliche in modo uniforme su tutti gli shard del cluster.

ElastiCache con Valkey o Redis OSS 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 l'applicazione ElastiCache con scalabilità OSS automatica Valkey o Redis](#): 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 ridimensionamento OSS automatico ElastiCache con Valkey o Redis come mostrato nel diagramma precedente:

1. È possibile creare una policy di scalabilità ElastiCache automatica per il gruppo di replica.
2. ElastiCache il ridimensionamento automatico con Valkey o Redis OSS 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 dell'obiettivo) per un periodo di tempo specifico, CloudWatch attiva un allarme che richiama l'auto scaling per valutare la politica di scalabilità.
4. ElastiCache con Valkey o Redis OSS auto scaling emette una richiesta di modifica per regolare la capacità del cluster.

5. ElastiCache con Valkey o Redis OSS 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 funziona ElastiCache con Valkey o Redis Auto OSS 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. Decidi anche un valore di destinazione per l'CPU utilizzo per il cluster. ElastiCache 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 con Valkey o Redis Auto OSS 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 cerca di mantenere l'utilizzo del target pari o vicino al valore prescelto a lungo termine.

## Policy di Auto Scaling

Una policy di dimensionamento dispone dei seguenti componenti:

- Una metrica target: la CloudWatch metrica utilizzata ElastiCache con Valkey o Redis Auto OSS Scaling per determinare quando e quanto scalare.
- 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 attuale è superiore alla capacità massima configurata, lo facciamo `scaleIn` durante la creazione della policy. Allo stesso modo, se la capacità attuale è inferiore alla capacità minima configurata, procediamo `scaleOut` al `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 AWS Identity and Access Management (IAM) collegato a un servizio specifico AWS . Un ruolo collegato al servizio include tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per conto dell'utente. ElastiCache con Valkey o Redis OSS Auto Scaling genera automaticamente 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, una metrica predefinita o personalizzata e un valore target per la metrica vengono specificati in una configurazione della policy di target-tracking scaling. ElastiCache con Valkey o Redis Auto OSS Scaling crea e CloudWatch gestisce allarmi che attivano la politica di scalabilità e calcola la regolazione della scalabilità in base alla metrica e al valore target. 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 politica può mantenere l'CPUutilizzo pari o vicino a una percentuale di utilizzo specificata, ad esempio il 70 per cento.

 Note

Per ogni cluster , è possibile creare solo una policy di Auto Scaling per ogni parametro di destinazione.

## Capacità minima e massima


### Shard

È possibile specificare il numero massimo di shard a cui è possibile scalare ElastiCache con l'auto scaling Valkey o Redis. OSS Questo valore deve essere uguale o inferiore a 250 con un minimo di 1. È inoltre possibile specificare il numero minimo di shard da gestire mediante la scalabilità automatica. 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 da gestire ElastiCache con la scalabilità OSS automatica Valkey o Redis. Questo valore deve essere maggiore o uguale a 5. È inoltre possibile specificare il numero minimo di repliche da gestire mediante la scalabilità automatica. 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

ElastiCache con le politiche di OSS auto scaling di Valkey o Redis, aumenta la capacità del cluster fino a raggiungere la dimensione massima definita o fino all'applicazione dei limiti di servizio. Per richiedere un aumento dei limiti, consulta [AWS Limiti 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 con Valkey o Redis si ridimensiona OSS automaticamente fino 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. Ciò rallenta l'eliminazione di shard/repliche nel OSS cluster ElastiCache with Valkey o Redis per le richieste di scalabilità orizzontale e la creazione di shard/repliche per le richieste di scalabilità orizzontale. Puoi specificare i seguenti periodi di attesa:

- Un'attività di scalabilità riduce il numero di shard/repliche nel cluster. 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 scalabilità orizzontale aumenta il numero di shard/repliche nel cluster. 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 scalabilità orizzontale sono sempre abilitate in modo che la policy di scalabilità possa creare shard/repliche Valkey o Redis in base alle esigenze. ElastiCache OSS

## IAMAutorizzazioni richieste per Auto Scaling

ElastiCache con Valkey o Redis Auto OSS Scaling è reso possibile dalla combinazione di ElastiCache CloudWatch, e Application Auto Scaling. APIs I cluster vengono creati e aggiornati

con ElastiCache (RedisOSS), gli allarmi vengono creati e le politiche di scalabilità vengono create con CloudWatch Application Auto Scaling. Oltre alle IAM autorizzazioni standard per la creazione e l'aggiornamento dei cluster, l'IAMutente che accede alle impostazioni di ElastiCache Auto Scaling deve disporre delle autorizzazioni appropriate per i servizi che supportano la scalabilità dinamica. IAMgli utenti devono disporre delle autorizzazioni per utilizzare le azioni mostrate nella seguente politica 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 OSS auto scaling ElastiCache con Valkey o Redis necessita inoltre dell'autorizzazione per descrivere i cluster e gli CloudWatch allarmi e delle autorizzazioni per modificare la capacità target per tuo ElastiCache conto. Se abiliti Auto Scaling per il tuo cluster, viene creato un ruolo collegato al servizio denominato.

`AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG` Questo ruolo collegato al servizio concede l'autorizzazione alla scalabilità ElastiCache automatica per descrivere gli allarmi per le tue politiche, monitorare la capacità attuale del parco veicoli e modificare la capacità del parco veicoli. Il ruolo collegato al servizio è il ruolo predefinito per la scalabilità ElastiCache automatica. Per ulteriori informazioni, consulta [Service-linked roles for ElastiCache \(Redis\) OSS auto scaling 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. Utilizza solo una metrica di tracciamento: identifica se il tuo cluster presenta CPU carichi di lavoro a uso intensivo di dati e utilizza una metrica predefinita corrispondente per definire la politica di scalabilità.
  - MotoreCPU: `ElastiCachePrimaryEngineCPUUtilization` (dimensione del frammento) o (dimensione della replica) `ElastiCacheReplicaEngineCPUUtilization`
  - Utilizzo del database:  
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` questa policy di dimensionamento è ideale con `maxmemory-policy` impostato su `noeviction` sul cluster.

Ti consigliamo di evitare più politiche per dimensione nel cluster. ElastiCache con Valkey o Redis OSS Auto scaling ridimensionerà la destinazione scalabile se alcune policy di tracciamento del target sono pronte per la scalabilità orizzontale, ma la scalerà solo se tutte le politiche di tracciamento del target (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 utilizza Valkey o Redis OSS 7.0 o versioni successive e l'applicazione utilizza [Functions](#), [consigliamo di caricare tutte le funzioni](#) su tutti gli shard prima di eseguire la scalabilità orizzontale, in modo che il cluster non finisca con funzioni diverse su shard diversi.

Dopo la registrazione a, tieni presente quanto segue: AutoScaling

- 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

Con ElastiCache s AutoScaling puoi utilizzare le politiche di tracciamento e pianificazione con il tuo motore Valkey o RedisOSS.

Di seguito vengono forniti dettagli sul tracciamento degli obiettivi e sulle politiche pianificate e su come applicarle utilizzando e. AWS Management Console AWS CLI APIs

### Argomenti

- [Policy di dimensionamento con monitoraggio degli obiettivi](#)
- [Aggiunta di una policy di dimensionamento](#)
- [Registrare un target scalabile](#)
- [Definizione di una policy di dimensionamento](#)
- [Disabilitazione dell'attività di riduzione orizzontale](#)
- [Applicazione di una policy di dimensionamento](#)

- [Modifica di una policy di dimensionamento](#)
- [Eliminazione di una policy di dimensionamento](#)
- [Utilizzo AWS CloudFormation per le politiche di Auto Scaling](#)
- [Dimensionamento programmato](#)

## 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 con Valkey o Redis Auto OSS Scaling crea e gestisce gli allarmi che attivano CloudWatch la politica di scalabilità e calcola la regolazione della scalabilità in base alla metrica e al valore target. 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 `ElastiCachePrimaryEngineCPUUtilization` metrica con il valore di destinazione configurato. Tale politica può mantenere l'CPU utilizzo pari o vicino al valore target 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                                           | CloudWatch Nome della metrica                             | CloudWatch Dimensione metrica                          | Tipi di istanza non idonei |
|----------------------------------------------------------------------|-----------------------------------------------------------|--------------------------------------------------------|----------------------------|
| <code>ElastiCachePrimaryEngineCPUUtilization</code>                  | <code>EngineCPUUtilization</code>                         | Replicati<br><code>onGroupId</code> , Ruolo = primario | Nessuno                    |
| <code>ElastiCacheDatabaseCapacityUsageCountForEvictPercentage</code> | <code>DatabaseCapacityUsageCountForEvictPercentage</code> | Metriche del gruppo di OSS                             | Nessuno                    |



| Nome parametro predefinito                              | CloudWatch Nome della metrica                | CloudWatch Dimensione metrica                     | Tipi di istanza non idonei |
|---------------------------------------------------------|----------------------------------------------|---------------------------------------------------|----------------------------|
| eCapacity<br>UsageCountedForEvictPercentage             |                                              | replica Valkey o Redis                            |                            |
| ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage | DatabaseMemoryUsageCountedForEvictPercentage | Metriche del gruppo di replica Valkey o Redis OSS | R6gd                       |

I tipi di istanze a più livelli di dati non possono essere

utilizzati `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`, poiché questi tipi di istanza archiviano i dati sia in memoria che in SSD. Il caso d'uso previsto per le istanze a più livelli di dati consiste nell'utilizzare la memoria al 100% e riempirla secondo necessità. SSD

### 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 con Valkey o Redis, OSS ridimensiona gli shard del cluster in base a un conteggio pari al più grande tra due numeri: variazione percentuale rispetto a Target e 20% rispetto agli shard attuali. Per quanto riguarda la scalabilità in entrata, ElastiCache non verrà eseguita automaticamente a meno che il valore della metrica complessivo non sia inferiore al 75 per cento del Target definito.

Per un esempio di scala orizzontale, se hai 50 frammenti e

- se Target viene violato del 30 per cento, ElastiCache con Valkey o Redis la OSS scalabilità orizzontale si riduce del 30 per cento, il che si traduce in 65 shard per cluster.
- se Target viene violato del 10 per cento, ElastiCache con Valkey o Redis viene OSS ridimensionato di default almeno del 20 per cento, il che si traduce in 60 shard per cluster.

Ad esempio, se hai selezionato un valore Target del 60 percento, ElastiCache con Valkey o Redis non verrà eseguito lo scaling automatico finché la metrica OSS non sarà inferiore o uguale al 45 percento (25 percento inferiore al 60 percento di Target).

## 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 è possibile utilizzare una politica di ridimensionamento del tracciamento del target per la scalabilità orizzontale quando la metrica specificata è inferiore al valore target. ElastiCache con Valkey o Redis, OSS ridimensiona gli shard con una deviazione minima del 20% rispetto all'obiettivo degli shard 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. Questo perché ElastiCache con Valkey o Redis Auto OSS Scaling agisce sempre in modo conservativo arrotondando verso l'alto o verso il basso quando determina quanta capacità 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 disporre di più politiche di scalabilità di tracciamento degli obiettivi per un OSS cluster ElastiCache with Valkey o Redis, a condizione che ciascuna di esse utilizzi una metrica diversa. L'intenzione di ElastiCache (RedisOSS) Auto Scaling è quella di dare sempre la priorità alla disponibilità, quindi il suo comportamento varia a seconda che le policy di tracciamento di destinazione siano pronte per la scalabilità orizzontale o verticale. 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 modificate o eliminate gli CloudWatch allarmi che ElastiCache con Valkey o Redis OSS Auto Scaling gestiscono una politica di scalabilità di tracciamento degli obiettivi. ElastiCache Auto Scaling elimina automaticamente gli allarmi quando si elimina la politica di ridimensionamento.

- ElastiCache L'Auto Scaling non impedisce di modificare manualmente gli shard del cluster. Queste regolazioni manuali non influiscono sugli CloudWatch allarmi esistenti associati alla politica di scalabilità, ma possono influire sulle metriche che potrebbero attivare questi allarmi. CloudWatch
- Questi CloudWatch allarmi gestiti da Auto Scaling sono definiti sulla base AVG della metrica su tutti gli shard del cluster. Quindi, avere frammenti caldi può comportare uno scenario di:
  - ridimensionamento quando non necessario a causa del caricamento su alcuni hot shard che fa scattare un allarme CloudWatch
  - non si ridimensiona quando necessario a causa dell'aggregazione di AVG tutti i frammenti che influiscono sull'allarme da non violare.
- ElastiCache con Valkey o Redis si applicano ancora i limiti OSS predefiniti 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](#) scegliere il tipo di limiteNodi per cluster per tipo di istanza.
- Assicurati di avere abbastanza interfacce di rete ENIs (Elastic Network Interface) disponibili nel tuo dispositivoVPC, necessarie durante la scalabilità orizzontale. Per ulteriori informazioni, consulta [Interfacce di rete elastiche](#).
- Se non è disponibile una capacità sufficienteEC2, ElastiCache Auto Scaling non scalerà e verrà ritardato fino a quando la capacità non sarà disponibile.
- ElastiCache (RedisOSS) Auto Scaling durante lo scale-in non rimuoverà gli shard con slot con una dimensione dell'elemento superiore a 256 MB dopo la 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

È possibile aggiungere una politica di scalabilità utilizzando. AWS Management Console

Per aggiungere una policy di Auto Scaling a un cluster ElastiCache con Valkey o Redis OSS

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS.
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:
    - CPUUtilizzo primario per creare una politica basata sull'utilizzo medio. 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 della capacità include la memoria e l'SSDutilizzo per le istanze a più livelli di dati e l'utilizzo della memoria per tutti gli altri tipi di istanze.
  9. Per il valore target, scegli un valore maggiore o uguale a 35 e inferiore o uguale a 70. La scalabilità automatica manterrà questo valore per la metrica di destinazione selezionata tra i tuoi shard: ElastiCache
    - CPUUtilizzo primario: mantiene il valore target per la EngineCPUUtilization metrica 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. Utilizzate il AWS CLI per modificare i valori di cooldown.
  11. Per Capacità minima, digita il numero minimo di shard che la policy ElastiCache Auto Scaling deve mantenere.
  12. Per Capacità massima, digitare il numero massimo di shard che la policy ElastiCache Auto Scaling deve mantenere. Questo valore deve essere maggiore o uguale a 250.
  13. Scegli Create (Crea) .

## Registrare un target scalabile

Prima di poter utilizzare Auto Scaling con un cluster ElastiCache con Valkey o Redis, è necessario registrare il OSS cluster con auto scaling. ElastiCache Lo fai per definire la dimensione di scalabilità e

i limiti da applicare a quel cluster. ElastiCache la scalabilità automatica ridimensiona dinamicamente il cluster lungo la dimensione `elasticache:replication-group:NodeGroups` scalabile, che rappresenta il numero di shard del cluster.

Utilizzando il AWS CLI

Per registrarti ElastiCache con il OSS cluster Valkey o Redis, usa il [register-scalable-target](#) comando con i seguenti parametri:

- `--service-namespace` – Imposta questo valore su `elasticache`
- `--resource-id`— L'identificatore di risorsa per il cluster. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, ad esempio. `replication-group/myscalablecluster`
- `--scalable-dimension`: imposta questo valore su `elasticache:replication-group:NodeGroups`.
- `--max-capacity` — Il numero massimo di shard da gestire mediante la scalabilità ElastiCache automatica. 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 shard da gestire tramite ElastiCache auto scaling. 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, si registra un cluster ElastiCache con Valkey o Redis OSS 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 ^
```

## Utilizzando il 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, ad esempio. `replication-group/myscalablecluster`
- **ScalableDimension** — Imposta questo valore su. `elasticache:replication-group:NodeGroups`
- **MinCapacity** — Il numero minimo di shard da gestire tramite ElastiCache auto scaling. 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 shard da gestire mediante la scalabilità ElastiCache automatica. 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 OSS cluster ElastiCache with Valkey o Redis denominato `myscalablecluster` Application Auto Scaling. API 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:NodeGroups",
 "MinCapacity": 1,
 "MaxCapacity": 5
}
```

## Definizione di una policy di dimensionamento

Una configurazione della politica di scalabilità del target-tracking è rappresentata da un JSON blocco in cui sono definite le metriche e i valori target. È possibile salvare una configurazione della politica di scalabilità come JSON blocco in un file di testo. Si utilizza quel file di testo quando si richiama il AWS CLI o l'Application Auto API Scaling. Per ulteriori informazioni sulla sintassi di configurazione delle policy, vedere [TargetTrackingScalingPolicyConfiguration](#) Application Auto API Scaling 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](#)

### Utilizzo di un parametro di default

Utilizzando metriche predefinite, puoi definire rapidamente una politica di scalabilità del tracciamento del target per un OSS cluster ElastiCache con Valkey o Redis che funziona con il tracciamento del target in (Redis) Auto Scaling. ElastiCache OSS

Attualmente, ElastiCache supporta le seguenti metriche predefinite in Auto NodeGroup Scaling:

- `ElastiCachePrimaryEngineCPUUtilization`— Il valore medio della `EngineCPUUtilization` metrica in tutti i nodi primari del CloudWatch cluster.
- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`— Il valore medio della `DatabaseMemoryUsageCountedForEvictPercentage` metrica in CloudWatch tutti i nodi primari del cluster.

- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`— Il valore medio della `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` metrica in CloudWatch tutti i nodi primari del cluster.

Per ulteriori informazioni sulle metriche `EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` e `DatabaseCapacityUsageCountedForEvictPercentage`, consultare [Monitoraggio dell'utilizzo con CloudWatch Metrics](#). 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 ElastiCache cluster with Valkey o Redis. OSS In questa configurazione, la metrica `ElastiCachePrimaryEngineCPUUtilization` predefinita viene utilizzata per regolare il cluster in base a un CPU utilizzo medio 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 cluster ElastiCache (RedisOSS) in base a un CPU utilizzo medio 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, vedere [TargetTrackingScalingPolicyConfiguration](#) 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 (RedisOSS) in base a un CPU utilizzo medio 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 la scalabilità della configurazione della policy di scalabilità target-tracking nel cluster disabilitando l'attività di scalabilità in ingresso. 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, vedere [TargetTrackingScalingPolicyConfiguration](#) 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 ElastiCache con Valkey o Redis in base a un CPU utilizzo medio del 40% su tutti i nodi primari di quel OSS 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 ElastiCache con Valkey o Redis OSS auto scaling e aver definito una politica di scalabilità, applichi la politica di scalabilità al cluster registrato. Per applicare una politica di

scalabilità a un cluster ElastiCache (RedisOSS), puoi utilizzare AWS CLI o Application Auto Scaling API

Applicazione di una politica di scalabilità utilizzando AWS CLI

Per applicare una politica di scalabilità al tuo OSS cluster ElastiCache with Valkey o Redis, usa 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 della risorsa. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, ad esempio. `replication-group/myscalablecluster`
- `—service-namespace`— Impostare questo valore su `elasticache`.
- `—scalable-dimension`— Impostare questo valore su `elasticache:replication-group:NodeGroups`.
- `--target-tracking-scaling-policy-configuration` — La configurazione della policy di scalabilità target-tracking da utilizzare per il cluster.

Nell'esempio seguente, si applica una politica di scalabilità di tracciamento della destinazione denominata `myscalablepolicy` a un cluster ElastiCache with Valkey o Redis OSS denominato `myscalablecluster` 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 politica di scalabilità utilizzando API

Per applicare una politica di scalabilità al tuo OSS cluster ElastiCache with Valkey o Redis, usa il [PutScalingPolicy](#) AWS CLI comando con i seguenti parametri:

- `—policy-name—` Il nome della policy di dimensionamento.
- `--resource-id —` L'identificatore della risorsa. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, ad esempio. `replication-group/myscalablecluster`
- `—service-namespace—` Impostare questo valore su `elasticache`.
- `—scalable-dimension—` Impostare questo valore su `elasticache:replication-group:NodeGroups`.
- `-- target-tracking-scaling-policy -configuration —` La configurazione della policy di scalabilità target-tracking da utilizzare per il cluster.

Nell'esempio seguente, si applica una politica di scalabilità di tracciamento della destinazione denominata `myscalablepolicy` a un cluster ElastiCache with Valkey o Redis OSS denominato con `auto scaling. myscalablecluster` 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": "elasticache",
```

```
"ResourceId": "replication-group/myscalablecluster",
"ScalableDimension": "elasticache:replication-group:NodeGroups",
"PolicyType": "TargetTrackingScaling",
"TargetTrackingScalingPolicyConfiguration": {
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 }
}
```

## Modifica di una policy di dimensionamento

È possibile modificare una politica di scalabilità utilizzando Application Auto Scaling AWS CLI, o Application Auto API Scaling. AWS Management Console

## Modifica di una politica di scalabilità utilizzando AWS Management Console

Per modificare una policy di Auto Scaling per un cluster ElastiCache con Valkey o Redis OSS

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli il motore appropriato.
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 politica di scalabilità utilizzando e AWS CLI API

È possibile utilizzare AWS CLI o Application Auto Scaling API per modificare una policy di scaling nello stesso modo in cui si applica una policy di scaling:

- Quando si utilizza il AWS CLI, specificare il nome della politica che si desidera modificare nel parametro. `--policy-name` Specifica i nuovi valori per i parametri che desideri modificare.

- Quando si utilizza Application Auto Scaling API, specificare il nome della politica che si desidera modificare nel `PolicyName` parametro. 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 politica di scalabilità utilizzando il AWS Management Console AWS CLI, o Application Auto API Scaling.

### Eliminazione di una politica di scalabilità utilizzando il AWS Management Console

Per eliminare una policy di Auto Scaling per un cluster ElastiCache (RedisOSS)

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS.
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 politica di scalabilità utilizzando il AWS CLI

Per eliminare una politica di scalabilità in base al tuo OSS cluster ElastiCache con Valkey o Redis, usa il [delete-scaling-policy](#) AWS CLI comando con i seguenti parametri:

- `--policy-name`— Il nome della policy di dimensionamento.
- `--resource-id` — L'identificatore della risorsa. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, 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, si elimina una politica di scalabilità di tracciamento degli obiettivi denominata `myscalablepolicy` da un cluster 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 politica di scalabilità utilizzando il API

Per eliminare una politica di scalabilità in base al tuo OSS cluster ElastiCache con Valkey o Redis, usa il [DeleteScalingPolicy](#) AWS CLI comando con i seguenti parametri:

- `--policy-name`— Il nome della policy di dimensionamento.
- `--resource-id` — L'identificatore della risorsa. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, 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, si elimina una politica di scalabilità di tracciamento degli obiettivi denominata `myscalablepolicy` da un cluster 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"
}

```

## Utilizzo AWS CloudFormation per le politiche di Auto Scaling

Questo frammento mostra come creare una politica di tracciamento degli obiettivi e applicarla a una risorsa [AWS::ElastiCache](#): utilizzando la ReplicationGroup 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 ridimensionamento pianificato, si creano azioni pianificate, che indicano ElastiCache a Valkey o Redis di OSS eseguire attività di scalabilità in momenti specifici. Quando si crea un'azione pianificata, si specifica un cluster esistente ElastiCache (RedisOSS), quando deve avvenire l'attività di scalabilità, la capacità minima e la capacità massima. È possibile creare operazioni pianificate sia una tantum che ricorrenti.

È possibile creare un'azione pianificata solo per i cluster ElastiCache (RedisOSS) 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 a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS.
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 a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS.
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 a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS.
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

Utilizzate la seguente scalabilità automatica delle applicazioni: APIs

- [put-scheduled-action](#)

- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Utilizza AWS CloudFormation per creare un'operazione pianificata

Questo frammento mostra come creare una politica di tracciamento degli obiettivi e applicarla a una risorsa [AWS::ElastiCache::](#) utilizzando la ReplicationGroup 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

Un gruppo di ElastiCache replica può configurare una o più cache per funzionare come un singolo nodo logico.

Di seguito vengono forniti dettagli sul tracciamento degli obiettivi e sulle politiche pianificate e su come applicarle utilizzando and. AWS Management Console AWS CLI APIs

## 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 con Valkey o Redis OSS AutoScaling crea e gestisce gli CloudWatch allarmi che attivano la politica di scalabilità e calcola la regolazione della scalabilità in base alla metrica e al valore target. 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 CPU utilizzo AVG del motore aggregata tra tutte le repliche utilizzata per attivare un' ElastiCache operazione di auto-scaling. È possibile impostare l'obiettivo di utilizzo tra il 35 per cento e il 70 per cento.

Quando il servizio rileva che la `ElastiCacheReplicaEngineCPUUtilization` metrica è uguale o superiore all'impostazione di Target, aumenterà automaticamente le repliche tra gli shard. ElastiCache con Valkey o Redis, OSS ridimensiona le repliche del cluster in base a un conteggio pari al più grande tra due numeri: variazione percentuale rispetto a Target e una replica. Per quanto riguarda lo scale-in, ElastiCache con Valkey o Redis OSS non è possibile eseguire la scalabilità automatica a meno che il valore complessivo della metrica non sia inferiore al 75 per cento del Target definito.

Per un esempio di aumento orizzontale, se si dispone di 5 shard e 1 replica ciascuno:

Se Target viola del 30 per cento, ElastiCache con Valkey o Redis viene OSS ridimensionato di 1 replica ( $\max(0,3, \text{impostazione predefinita } 1)$ ) su tutti gli shard, il che si traduce in 5 shard con 2 repliche ciascuno,

Ad esempio, se hai selezionato un valore Target del 60 per cento, ElastiCache con Valkey o Redis non verrà eseguito lo scaling automatico finché la metrica OSS non sarà inferiore o uguale al 45 per cento (25 per cento inferiore al 60 per cento di Target).

### 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 è possibile utilizzare una politica di ridimensionamento del tracciamento del target per eseguire la scalabilità orizzontale quando la metrica specificata è inferiore al valore target. ElastiCache con Valkey o Redis, OSS ridimensiona le repliche in base al massimo (deviazione in% arrotondata rispetto a Target, impostazione predefinita 1) delle repliche esistenti su tutti gli shard 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. Questo perché ElastiCache con Valkey o Redis Auto OSS Scaling agisce sempre in modo conservativo arrotondando verso l'alto o verso il basso quando determina quanta capacità 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 disporre di più politiche di scalabilità di tracciamento degli obiettivi per un OSS cluster ElastiCache with Valkey o Redis, a condizione che ciascuna di esse utilizzi una metrica diversa. L'intenzione di Auto Scaling è quella di dare sempre la priorità alla disponibilità, quindi il suo comportamento varia a seconda che le policy di tracciamento di destinazione siano pronte per la scalabilità orizzontale o verticale. 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 modificate o eliminate gli CloudWatch allarmi che ElastiCache con Valkey o Redis OSS Auto Scaling gestiscono una politica di scalabilità di tracciamento degli obiettivi. Auto Scaling elimina automaticamente gli allarmi quando si elimina la politica di scalabilità o si elimina il cluster.
- ElastiCache con Valkey o Redis Auto OSS Scaling non impedisce di modificare manualmente le repliche tra gli shard. Queste regolazioni manuali non influiscono sugli CloudWatch allarmi esistenti associati alla politica di scalabilità, ma possono influire sulle metriche che possono attivare questi allarmi. CloudWatch
- Questi CloudWatch allarmi gestiti da Auto Scaling sono definiti sulla base AVG della metrica su tutti gli shard del cluster. Quindi, avere frammenti caldi può comportare uno scenario di:
  - ridimensionamento quando non necessario a causa del caricamento su alcuni hot shard che fa scattare un allarme CloudWatch

- non si ridimensiona quando necessario a causa dell'aggregazione di AVG tutti i frammenti che influiscono sull'allarme da non violare.
- ElastiCache con Valkey o Redis si applicano ancora i limiti OSS predefiniti 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](#) scegliere il tipo di limite Nodi per cluster per tipo di istanza.
- Assicurati di avere abbastanza interfacce di rete ENIs (Elastic Network Interface) disponibili nel tuo dispositivo VPC, necessarie durante la scalabilità orizzontale. Per ulteriori informazioni, consulta [Interfacce di rete elastiche](#).
- Se non è disponibile una capacità sufficiente EC2, ElastiCache con Valkey o Redis Auto OSS Scaling non si esegue la scalabilità orizzontale fino a quando la capacità non è disponibile o se si modifica manualmente il cluster in base ai tipi di istanze con capacità sufficiente.
- ElastiCache con Valkey o Redis Auto OSS Scaling non supporta la scalabilità delle repliche con un cluster ReservedMemoryPercent con meno del 25 per cento. Per ulteriori informazioni, consulta [Gestione della memoria riservata per Valkey e Redis OSS](#).

## 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 scalabilità automatica ElastiCache con Valkey o Redis OSS

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS.
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 di destinazione, digitate la percentuale media di CPU utilizzo che desiderate mantenere sulle repliche. ElastiCache Questo valore deve essere tra  $\geq 35$  e  $\leq 70$ . 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 with Valkey o Redis OSS Auto Scaling deve gestire.
12. Per Capacità massima, digitare il numero massimo di repliche che la politica ElastiCache con Valkey o Redis OSS Auto Scaling deve mantenere. Questo valore deve essere  $\geq 5$ .
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'Application Auto ScalingAPI. Il primo passo è registrare il proprio gruppo di OSS replica ElastiCache con Valkey o Redis con Auto Scaling.

Prima di poter utilizzare la scalabilità ElastiCache automatica con un cluster, è necessario registrare il cluster ElastiCache con Valkey o Redis OSS auto scaling. Lo fai per definire la dimensione di scalabilità e i limiti da applicare a quel cluster. ElastiCache con Valkey o Redis OSS auto scaling ridimensiona dinamicamente il cluster lungo la dimensione `elasticache:replication-group:Replicas` scalabile, che rappresenta il numero di repliche del cluster per shard.

## Usando il CLI

Per registrare il ElastiCache cluster, utilizzate il [register-scalable-target](#) comando con i seguenti parametri:

- `—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, ad esempio. `replication-group/myscalablecluster`
- `—scalable-dimension` — Imposta questo valore su `elasticache:replication-group:Replicas`.
- `--min-capacity` — Il numero minimo di repliche da gestire ElastiCache con la scalabilità automatica Valkey o Redis. OSS 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 ElastiCache con la scalabilità automatica Valkey o Redis. OSS 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 ElastiCache cluster con Valkey o Redis denominato. OSS `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 \

```

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 ^\

```

## Usando il 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, ad esempio. `replication-group/myscalablecluster`
- `ScalableDimension` — Imposta questo valore su. `elasticache:replication-group:Replicas`



- **MinCapacity** — Il numero minimo di repliche da gestire ElastiCache con la scalabilità OSS automatica Valkey o 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 gestire ElastiCache con la scalabilità OSS automatica Valkey o 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 denominato `myscalablecluster` Application Auto Scaling API. 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 politica di scalabilità del target-tracking è rappresentata da un JSON blocco in cui sono definite le metriche e i valori target. È possibile salvare una configurazione della politica di scalabilità come JSON blocco in un file di testo. Si utilizza quel file di testo quando si richiama il AWS CLI o l'Application Auto API Scaling. Per ulteriori informazioni sulla sintassi di configurazione delle policy, vedere [TargetTrackingScalingPolicyConfiguration](#) Application Auto API Scaling 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](#)
- [Modifica di una policy di dimensionamento](#)
- [Eliminazione di una policy di dimensionamento](#)
- [Utilizzo AWS CloudFormation per le politiche di Auto Scaling](#)
- [Dimensionamento programmato](#)

### Utilizzo di un parametro di default

Una configurazione di policy di scalabilità con tracciamento degli obiettivi è rappresentata da un JSON blocco in cui sono definite le metriche e i valori target. È possibile salvare una configurazione della politica di scalabilità come JSON blocco in un file di testo. Si utilizza quel file di testo quando si richiama il AWS CLI o l'Application Auto API Scaling. Per ulteriori informazioni sulla sintassi di configurazione delle policy, vedere [TargetTrackingScalingPolicyConfiguration](#) Application Auto API Scaling 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 politica di scalabilità a un cluster con Valkey o Redis ElastiCache OSS](#)

### Utilizzo di un parametro di default

Utilizzando metriche predefinite, puoi definire rapidamente una politica di scalabilità del tracciamento del target per un OSS cluster con Valkey o Redis che funziona ElastiCache con il tracciamento del target con Valkey o Redis Auto Scaling. ElastiCache OSS Attualmente, ElastiCache supporta la seguente metrica predefinita in ElastiCache Replicas Auto Scaling:

`ElastiCacheReplicaEngineCPUUtilization`— Il valore medio della `ngineCPUUtilization` metrica E in tutte le repliche del CloudWatch cluster. Puoi trovare il valore aggregato della metrica

## CloudWatch in basso ElastiCache ReplicationGroupId, Role per Required e Role Replica. ReplicationGroupId

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. Puoi definire una metrica personalizzata basata su qualsiasi metrica ElastiCache con Valkey o Redis che cambia in proporzione alla OSS 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 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, una metrica personalizzata regola un cluster in base a un CPU utilizzo medio del 50% su tutte le repliche 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": "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 del cluster. Per ulteriori informazioni su `ScaleInCooldown` e `ScaleOutCooldown`, vedere [TargetTrackingScalingPolicyConfiguration](#) 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 `ElastiCacheReplicaEngineCPUUtilization` predefinita viene utilizzata per regolare un cluster in base a un CPU utilizzo medio del 40% su tutte le repliche 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": "ElastiCacheReplicaEngineCPUUtilization"
 },
 "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 ElastiCache con Valkey o Redis disabilitando l'attività di scalabilità. OSS 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 ulteriori informazioni [TargetTrackingScalingPolicyConfiguration](#) in merito `DisableScaleIn`, vedere Application Auto Scaling API Reference.

## Example

Il seguente esempio descrive una configurazione di monitoraggio degli obiettivi per una policy di dimensionamento. In questa configurazione, la metrica `ElastiCacheReplicaEngineCPUUtilization` predefinita regola un cluster in base a un CPU

utilizzo medio del 40% su tutte le repliche di 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 politica di scalabilità a un cluster con Valkey o Redis ElastiCache OSS

Dopo aver registrato il cluster ElastiCache con Valkey o Redis OSS auto scaling e aver definito una politica di scalabilità, applichi la politica di scalabilità al cluster registrato. Per applicare una politica di scalabilità a un OSS cluster ElastiCache con Valkey o Redis, puoi utilizzare AWS CLI o Application Auto Scaling. API

### Utilizzando il AWS CLI

Per applicare una politica di scalabilità al tuo OSS cluster ElastiCache con Valkey o Redis, usa 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 il cluster. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, 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-policy -configuration` — La configurazione della policy di scalabilità target-tracking da utilizzare per il cluster.

### Example

Nell'esempio seguente, si applica una politica di scalabilità di tracciamento della destinazione denominata `myscalablepolicy` a un cluster denominato `myscalablecluster` ElastiCache con

Valkey o Redis auto scaling. OSS 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
```

Utilizzando il API

Per applicare una policy di scaling al tuo OSS cluster ElastiCache with Valkey o Redis con Application Auto API Scaling, usa l'operazione Application Auto [PutScalingPolicy](#)Scaling con i seguenti parametriAPI:

- PolicyName — Il nome della politica di scalabilità.
- PolicyType — Imposta questo valore su. TargetTrackingScaling

- `resourceID`: l'identificatore della risorsa per il cluster. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster ElastiCache (RedisOSS), ad esempio `replication-group/myscalablecluster`
- `ServiceNamespace` — Imposta questo valore su `elasticache`.
- `ScalableDimension` — Imposta questo valore su `elasticache:replication-group:Replicas`
- `TargetTrackingScalingPolicyConfiguration` — La configurazione della politica di scalabilità del `target-tracking` da utilizzare per il cluster.

## Example

Nell'esempio seguente, si applica una politica di scalabilità di tracciamento della destinazione denominata `scalablepolicy` a un cluster denominato `myscalablecluster` ElastiCache con Valkey o Redis auto scaling. OSS 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 politica di scalabilità utilizzando il AWS Management Console AWS CLI, o Application Auto API Scaling.

## Modifica di una politica di scalabilità utilizzando il AWS Management Console

È possibile modificare i criteri solo con tipo Parametri predefiniti utilizzando la AWS Management Console

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS
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 Modify (Modifica).
6. Apportare le modifiche necessarie alla policy.
7. Scegli Modifica.
8. Apportare modifiche alla policy.
9. Scegli Modifica.

## Modifica di una politica di scalabilità utilizzando AWS CLI o l'Application Auto Scaling API

È possibile utilizzare AWS CLI o Application Auto Scaling API per modificare una policy di scaling nello stesso modo in cui si applica una policy di scaling:

- Quando si utilizza Application Auto ScalingAPI, specificare il nome della politica che si desidera modificare nel PolicyName parametro. Specifica i nuovi valori per i parametri che desideri modificare.

Per ulteriori informazioni, consulta [Applicazione di una politica di scalabilità a un cluster con Valkey o Redis ElastiCache OSS](#).



## Eliminazione di una policy di dimensionamento

È possibile eliminare una politica di scalabilità utilizzando il AWS Management Console, il AWS CLI o l'Application Auto Scaling API

### Eliminazione di una politica di scalabilità utilizzando il AWS Management Console

È possibile modificare i criteri solo con tipo Parametri predefiniti utilizzando la AWS Management Console

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS
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 AWS CLI o l'Application Auto Scaling API

È possibile utilizzare AWS CLI o l'Application Auto Scaling API per eliminare una politica di scalabilità da un cluster. ElastiCache

#### CLI

Per eliminare una politica di scalabilità dal tuo OSS cluster ElastiCache with Valkey o Redis, usa 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. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, ad esempio. `replication-group/myscalablecluster`
- `--service-namespace` — Impostare questo valore su `elasticache`.
- `--scalable-dimension` — Imposta questo valore su `elasticache:replication-group:Replicas`.

## Example

Nell'esempio seguente, si elimina una politica di scalabilità di tracciamento degli obiettivi denominata `myscalablepolicy` da un ELC; cluster 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 politica di scalabilità dal tuo OSS cluster ElastiCache con Valkey o Redis, utilizza l'operazione Application Auto [DeleteScalingPolicy](#) API Scaling con i seguenti parametri:

- `PolicyName` — Il nome della politica di scalabilità.
- `resourceID`: l'identificatore della risorsa per il cluster. Per questo parametro, il tipo di risorsa è `ReplicationGroup` e l'identificatore univoco è il nome del cluster, ad esempio. `replication-group/myscalablecluster`
- `ServiceNamespace` — Imposta questo valore su `elasticache`.
- `ScalableDimension` — Imposta questo valore su. `elasticache:replication-group:Replicas`

Nell'esempio seguente, si elimina una politica di scalabilità di tracciamento della destinazione denominata `myscalablepolicy` da un cluster denominato `myscalablecluster` Application Auto Scaling. API

```

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"
}

```

## Utilizzo AWS CloudFormation per le politiche di Auto Scaling

Questo frammento mostra come creare un'azione pianificata e applicarla a una risorsa [AWS::ElastiCache](#): utilizzando la ReplicationGroup 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 ridimensionamento pianificato, si creano azioni pianificate, che indicano ElastiCache a Valkey o Redis di OSS eseguire attività di scalabilità in momenti specifici. Quando si crea un'azione pianificata, si specifica un OSS cluster esistente ElastiCache con Valkey o Redis, quando deve avvenire l'attività di scalabilità, la capacità minima e la capacità massima. È possibile creare operazioni pianificate sia una tantum che ricorrenti.

È possibile creare un'azione pianificata solo per ElastiCache i cluster Valkey o Redis già esistenti OSS. 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, consulta [Dimensionamento programmato](#).

Per eliminare un'operazione pianificata

Simile alla dimensione Partizione. Per informazioni, consulta [Dimensionamento programmato](#).

Per gestire il dimensionamento pianificato tramite AWS CLI

Utilizza la seguente scalabilità automatica delle applicazioni: APIs

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

## Utilizzabile AWS CloudFormation per creare politiche di Auto Scaling

Questo frammento mostra come creare un'azione pianificata e applicarla a una risorsa [AWS::ElastiCache](#): utilizzando la ReplicationGroup 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

Valkey e Redis OSS sono database in memoria distribuiti che supportano lo sharding e la replica. ElastiCache I OSS cluster Valkey e Redis sono l'implementazione distribuita che consente il partizionamento dei dati su più nodi. Un cluster ElastiCache (RedisOSS) ha due modalità operative, la modalità Cluster abilitata (CME) e la modalità cluster disabilitata (). CMD InCME, un OSS motore Valkey e Redis funziona come un database distribuito con più shard e nodi, mentre inCMD, Valkey e Redis OSS funzionano come un singolo nodo.

Prima di effettuare la migrazione da CMD aCME, devono essere soddisfatte le seguenti condizioni:

**⚠ 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 OSS client Valkey o 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 richiesta per la migrazione è Valkey 7.2 e versioni successive o Redis 7.0 e versioni successive. OSS

Per migrare da CMD aCME, è necessario modificare la configurazione della modalità cluster da modalità cluster disabilitata a 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 (RedisOSS) sceglierà automaticamente il gruppo di parametri predefinito corrispondente con una configurazione abilitata per il cluster. Il valore del parametro abilitato al cluster è impostato su per un cluster. no 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 del motore utilizzando gruppi di ElastiCache parametri](#)

1. Preparazione: crea un CME cluster di test e assicurati che lo stack sia pronto per utilizzarlo. ElastiCache (RedisOSS) non ha modo di verificare la tua disponibilità. Per ulteriori informazioni, consulta [Creazione di un cluster per Valkey o Redis OSS](#).
2. Modifica la configurazione CMD del cluster esistente rendendola compatibile con la modalità cluster: in questa modalità, verrà distribuito un singolo shard e ElastiCache (RedisOSS) funzionerà come nodo singolo ma anche come cluster a singolo shard. 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 e l'endpoint di configurazione Valkey o Redis OSS Cluster. Per modificare la modalità cluster Valkey o Redis in una modalità compatibile con la modalità OSS cluster, procedi nel seguente modo:

#### 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 (`esclusicacheParameterGroupName`) non possono essere modificati quando si definisce il parametro in modalità cluster all'interno della richiesta.

[ModifyReplicationGroup](#)

- a. Utilizzando AWS Management Console, visualizza [Modifica di un gruppo di replica](#) e imposta la modalità cluster su `Compatibile`
- b. Utilizzando API, visualizza [ModifyReplicationGroup](#) e aggiorna il `ClusterMode` parametro su `compatible`.
- c. Utilizzando AWS CLI, visualizza [modify-replication-group](#) e aggiorna il `cluster-mode` parametro su `compatible`.

Dopo aver modificato la modalità OSS cluster Valkey o Redis in una modalità compatibile con la modalità cluster, [DescribeReplicationGroups](#) API restituirà l'endpoint di configurazione del cluster ElastiCache (RedisOSS). L'endpoint di configurazione del cluster è un endpoint singolo che può essere utilizzato dalle applicazioni per connettersi al cluster. Per ulteriori informazioni, consulta [Ricerca degli endpoint di connessione in ElastiCache](#).

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 tuoi OSS client Valkey o Redis siano migrati all'utilizzo del protocollo cluster e che l'endpoint di configurazione del cluster non sia in uso.

- a. Utilizzando AWS Management Console, visualizza [Modifica di un gruppo di replica](#) e imposta la modalità cluster su Enabled.
- b. Utilizzando API, visualizza [ModifyReplicationGroup](#) e aggiorna il `ClusterMode` parametro su `enabled`.
- c. Utilizzando AWS CLI, visualizza [modify-replication-group](#) e aggiorna il `cluster-mode` parametro su `enabled`.

Dopo aver modificato la modalità cluster su `enabled`, gli endpoint verranno configurati secondo le specifiche del cluster Valkey o RedisOSS. [DescribeReplicationGroups](#) API restituirà il parametro della modalità cluster `enabled` e gli endpoint del cluster che ora sono disponibili per essere utilizzati dalle applicazioni per la connessione 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 con la modalità cluster e conservare le configurazioni originali.

Modifica della configurazione del cluster nella modalità cluster disabilitata dalla modalità cluster compatibile

1. Utilizzando AWS Management Console, visualizza [Modifica di un gruppo di replica](#) e imposta la modalità cluster su Disabilitata
2. Utilizzando API, visualizza [ModifyReplicationGroup](#) e aggiorna il `ClusterMode` parametro su `disabled`.
3. Utilizzando AWS CLI, visualizza [modify-replication-group](#) e aggiorna il `cluster-mode` parametro su `disabled`.

Dopo aver modificato la modalità cluster in disabilitata, [DescribeReplicationGroups](#) API restituirà il parametro della modalità cluster `disabled`.



# Replica tra AWS regioni utilizzando datastore globali

## Note

Global Datastore è attualmente disponibile solo per i cluster progettati autonomamente.

Utilizzando la funzionalità Global Datastore, puoi lavorare con una replica di cluster Valkey o Redis completamente gestita, veloce, affidabile e sicura in tutte le regioni. OSS AWS Utilizzando questa funzionalità, è possibile creare cluster di replica di lettura interregionali per consentire letture a bassa latenza e disaster recovery tra le 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 datastore globali \(\) 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:

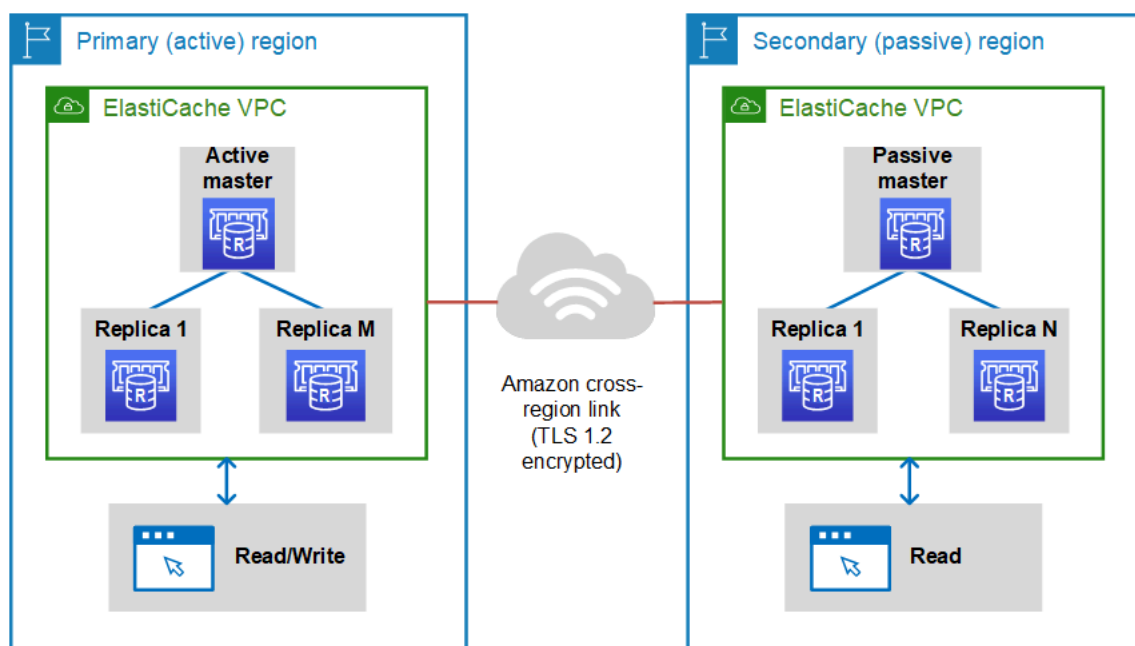
- Cluster primario (attivo)— Un cluster primario accetta scritture replicate in tutti i cluster all'interno del datastore globale. Un cluster primario accetta anche le richieste di lettura.
- Cluster secondario (passivo)— Un cluster secondario accetta solo richieste di lettura e replica gli aggiornamenti dei dati da un cluster primario. Un cluster secondario deve trovarsi in una regione diversa rispetto AWS al cluster primario.

Quando crei un datastore globale ElastiCache con Valkey o RedisOSS, esso replica automaticamente i dati dal cluster primario al cluster secondario. Scegli la AWS regione in cui i OSS dati Valkey o Redis devono essere replicati e quindi crei un cluster secondario in quella regione. AWS ElastiCache quindi configura e gestisce la replica automatica e asincrona dei dati tra i due cluster.

L'utilizzo di un datastore globale per Valkey o Redis offre i seguenti vantaggi: OSS

- Prestazioni geolocali: configurando cluster di replica remoti in AWS regioni aggiuntive e sincronizzando i dati tra di essi, è possibile ridurre la latenza di accesso ai dati in quella regione. AWS Un datastore globale può contribuire ad aumentare la reattività dell'applicazione fornendo letture geolocali a bassa latenza in tutte le 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. Puoi farlo connettendoti a qualsiasi regione che contenga un cluster secondario. AWS

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 sono supportati su istanze di dimensioni superiori o superiori.
- 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 lavorare con datastore globali solo in cluster. VPC Per ulteriori informazioni, consulta [Modelli di accesso per accedere a una ElastiCache cache in Amazon VPC](#). I datastore globali non sono supportati quando si utilizza -Classic. EC2 Per ulteriori informazioni, consulta [EC2-Classic](#) nella Amazon EC2 User Guide.

#### Note

Al momento, non è possibile utilizzare gli archivi dati globali in [Utilizzo delle zone locali 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 [Ridimensionamento ElastiCache](#).
- [I datastore globali supportano la crittografia a riposo, la crittografia in transito e. AUTH](#)

- I datastore globali non supportano la versione 6 del protocollo Internet (). 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 Valkey 7.2 o successivo oppure Redis OSS 5.0.6 o successivo.
2. Aggiungi fino a due cluster secondari in AWS regioni diverse, sempre utilizzando Valkey 7.2 o versione successiva oppure il motore Redis 5.0.6 o successivo. OSS

Le seguenti procedure ti guidano su come creare un datastore globale per Valkey o Redis OSS ed eseguire altre operazioni utilizzando la console. 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 un motore Valkey 7.2 o successivo o Redis 5.0.6 o successivo. OSS


## Per creare un Global Datastore utilizzando un cluster esistente

1. Accedi a AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione, scegli Datastore globale, quindi scegli Crea datastore globale.
3. Nella pagina delle impostazioni del cluster primario, procedi come segue:
  - Nel campo Informazioni Global Datastore, inserisci un nome per il nuovo datastore globale.
  - (Opzionale) Immettere un valore Description (Descrizione).
4. In Cluster regionale, seleziona Usa cluster regionale esistente.
5. In Cluster esistente, seleziona il cluster esistente che desideri utilizzare.
6. 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 gestita dal cliente e scegliendo AWS KMS la chiave. Per ulteriori informazioni, consulta [Utilizzo delle AWS KMS chiavi 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 Valkey 7.2 e versioni successive e il OSS motore Redis dalla versione 6.0 in poi, 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 la console e CLI](#).
    - AUTH utente predefinito: meccanismo di autenticazione per un server Valkey o Redis. OSS Per ulteriori informazioni, vedere. [AUTH](#)
7. (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 (AWS KMSchiave)
  - AUTHToken
  - Abilitazione backup automatici.
  - Backup retention period (Periodo di retention dei backup)
  - Finestra di backup
  - Maintenance window (Finestra di manutenzione)
  - Argomento per la SNS notifica
8. Scegli **Create (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 viene aggiornata per indicare se un cluster fa parte di un datastore globale, tra cui:

- **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 regione diversa AWS . 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 a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli **Datastore globale**, quindi scegli **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 Valkey e Redis OSS](#) e [Creazione di un gruppo di ElastiCache 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 [Suddivisione dei dati su più livelli in ElastiCache](#).

- e. Se stai creando un cluster Valkey o Redis OSS (modalità cluster disabilitata):

In Numero di repliche scegli il numero di repliche per questo cluster.

- f. Se stai creando un cluster Valkey o Redis OSS (modalità cluster abilitata):

- i. Per Numero di shard, scegli il numero di shard (partizioni/gruppi di nodi) che desideri per questo cluster Valkey o Redis (modalità cluster abilitata). OSS

Per alcune versioni di Valkey o Redis OSS (modalità cluster abilitata), puoi modificare il numero di shard nel cluster in modo dinamico:

- Redis OSS 3.2.10 e versioni successive: se sul cluster è in esecuzione Redis OSS 3.2.10 o versioni successive, è possibile modificare il numero di shard nel cluster in modo dinamico. Per ulteriori informazioni, consulta [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#).
  - Altre OSS versioni di Redis: se sul cluster è in esecuzione una versione di Redis precedente alla versione 3.2.10, esiste un altro approccio OSS. 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 Valkey o Redis OSS (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 ritieni di aver bisogno di più o meno repliche per shard (API/CLI: gruppo di nodi), devi

creare un nuovo cluster con il nuovo numero di repliche. Per ulteriori informazioni, consulta [Tutorial: seminare 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 IPv4 sottoreti predefinito oppure puoi scegliere di crearne uno nuovo. Infatti IPv6, è necessario creare un gruppo di sottoreti con un IPv6 CIDR blocco. Se si sceglie il dual stack, è necessario selezionare un tipo di Discovery IP, oppure IPv6. IPv4

Per ulteriori informazioni, consulta [Creare una sottorete in VPC](#).

7. Per Posizionamento zone disponibilità, le opzioni sono due:
  - Nessuna preferenza: ElastiCache seleziona 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 partizioni, seleziona la zona di disponibilità dall'elenco.


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

The screenshot shows the configuration interface for an ElastiCache cluster. At the top, there are two dropdown menus: "Slots and keyspaces" set to "Custom distribution" and "Availability zone(s)" set to "Specify availability zones". Below these is a table with three columns: "Slots/Keyspaces", "Primary", and "Replica 1". The table has three rows for "NodeGroup 1", "NodeGroup 2", and "NodeGroup 3". In the "Slots/Keyspaces" column, the value "0-1234" is circled in red. In the "Primary" column, the value "us-east-1b" is circled in red. The "Replica 1" column shows "us-east-1a" for all three node groups.

|             | Slots/Keyspaces | Primary    | Replica 1  |
|-------------|-----------------|------------|------------|
| NodeGroup 1 | 0-1234          | us-east-1a | us-east-1a |
| NodeGroup 2 |                 | us-east-1b | us-east-1a |
| NodeGroup 3 |                 | us-east-1a | us-east-1a |


Specifica dei Keyspace e delle zone di disponibilità

8. Scegli Successivo.
9. Nelle impostazioni Advanced Valkey e Redis OSS
  - 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 gestita dal cliente e scegliendo la AWS KMS chiave. Per ulteriori informazioni, consulta [Utilizzo delle chiavi gestite dai clienti](#) di AWS KMS.

- Crittografia dei dati in transito : Consente la crittografia dei dati in trasferimento. Per maggiori informazioni, consultare [Crittografia dei dati in transito](#). Per Valkey 7.2 e versioni successive e il OSS motore Redis versione 6.0 e successive, se abiliti Encryption in-transit ti verrà 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 la console e CLI](#).
  - AUTHUtente predefinito: meccanismo di autenticazione per un server Valkey o Redis. OSS Per ulteriori informazioni, vedere. [AUTH](#)
  - AUTH— Un meccanismo di autenticazione per un server Valkey o RedisOSS. Per ulteriori informazioni, vedere. [AUTH](#)

 Note

Per OSS le versioni Redis comprese tra 3.2.6 e successive, esclusa la versione 3.2.10, 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. Puoi utilizzare il gruppo di sicurezza predefinito per il tuo VPC o crearne uno nuovo.

Per ulteriori informazioni sui gruppi di sicurezza, consulta [la sezione Gruppi di sicurezza per te VPC](#) nella Amazon VPC User Guide.

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, consulta [Snapshot e ripristino](#).

11. (Facoltativo) Specifica una finestra di manutenzione. La finestra di manutenzione è l'orario, generalmente di un'ora, ogni settimana in cui si ElastiCache pianifica la manutenzione del sistema per il cluster. È possibile ElastiCache scegliere il giorno e l'ora della finestra di manutenzione (nessuna preferenza) oppure scegliere autonomamente il giorno, l'ora e la durata (Specificare la finestra di manutenzione). Se dagli elenchi si sceglie Specify maintenance window (Specificare 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 sono UCT orari.

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


12. (Facoltativo) PerRegistro:

- In Formato registro, scegli Testo 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 [Taggare le tue 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 di 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 gestita dal cliente e scegliendo AWS KMS la chiave. Per ulteriori informazioni, consulta [Utilizzo delle AWS KMS chiavi 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 Valkey 7.2 e versioni successive e il OSS motore Redis versione 6.4 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 la console e CLI](#).
- AUTHUtente predefinito: meccanismo di autenticazione per un server Valkey o Redis. OSS Per ulteriori informazioni, vedere. [AUTH](#)

#### Note

Per OSS le versioni Redis comprese tra la 4.0.2, quando Encryption in-transit era supportata per la prima volta, e la 6.0.4, 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
  - Customer Managed (chiave)AWS KMS
  - AUTHToken
  - Abilitazione backup automatici.
  - Backup retention period (Periodo di retention dei backup)
  - Finestra di backup
  - Maintenance window (Finestra di manutenzione)
  - Argomento per la SNS notifica
19. Scegli Create (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 viene inoltre aggiornata per indicare se un cluster fa parte di un datastore globale, tra cui:

- 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 regione diversa AWS . Per ulteriori informazioni, consulta [Aggiunta di una regione a un Global Datastore](#).

Visualizzazione dei dettagli del Datastore globale

È possibile visualizzare i dettagli dei datastore globali esistenti e modificarli anche nella pagina Global Datastores.

Per visualizzare i dettagli del Global Datastore

1. Accedi a AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione, scegli Datastore globali, quindi scegli un datastore globale 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

- Versione del motore: la versione del motore Valkey o Redis OSS che esegue il datastore globale
- 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
- AUTH: abilitato o disabilitato

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 Valkey o Redis abilitati alla modalità clusterOSS, il ritardo indica il ritardo massimo in secondi tra gli shard.



## 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 a AWS Management Console e apri la ElastiCache console all'indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione, scegli Datastore globali, quindi seleziona un datastore globale esistente.
3. Scegli Aggiungi cluster regionale e scegli la AWS regione in cui deve risiedere il cluster secondario.
4. In Informazioni sul cluster, inserisci un valore per Nome e, facoltativamente, per 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 gestita dal cliente e scegliendo AWS KMS la chiave.

- Crittografia in transito
  - 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
  - AWS KMSChiave gestita dal cliente)
  - AUTHToken
  - Abilitazione backup automatici.
  - Backup retention period (Periodo di retention dei backup)
  - Finestra di backup
  - Maintenance window (Finestra di manutenzione)
  - Argomento per la SNS notifica
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 a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Global Datastores, quindi per Global Datastore Name, scegli un datastore globale.
3. Scegliere Modifica e scegliere tra le seguenti opzioni:
  - Descrizione: una descrizione del Global Datastore
  - Modifica la versione del motore: sono disponibili solo Valkey 7.2 e versioni successive o il motore Redis OSS versione 5.0.6 e successive.
  - 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 Valkey o Redis con modalità cluster abilitata: OSS

- 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, usa la console Valkey o Redis OSS o l'operazione. [ModifyCacheCluster](#)API Per ulteriori informazioni, consulta [Modifica di un gruppo di ElastiCache 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, utilizzate l'[ResetCacheParameterGroupAPI](#) operazione.

## 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 con Valkey o Redis OSS non è garantito che alla fine quale diventi primario.

Per promuovere un cluster secondario a primario

1. Accedi a AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione, scegli Global Datastores.
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 a AWS Management Console e apri la ElastiCache console all'indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione, scegli Global Datastores.
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 a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel riquadro di navigazione, scegli Global Datastores.
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 datastore globali () CLI

È possibile utilizzare il AWS Command Line Interface (AWS CLI) per controllare più AWS servizi dalla riga di comando e automatizzarli tramite script. È possibile utilizzare il AWS CLI per operazioni ad hoc (una tantum).

### Scaricamento e configurazione di AWS CLI

AWS CLI Funziona su Windows, macOS o Linux. Utilizzare la procedura seguente per scaricarla e configurarla.

### Per scaricare, installare e configurare CLI

1. Scaricate la AWS CLI pagina Web dell'[interfaccia a AWS riga](#) di comando.
2. Segui le istruzioni per l'installazione AWS CLI e la configurazione riportate AWS CLI nella Guida per l'AWS Command Line Interface utente.

### Utilizzo di AWS CLI con datastore globali

Utilizzate le seguenti CLI operazioni per lavorare con datastore globali:

- [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 applica ElastiCache automaticamente un prefisso all'ID globale del datastore al momento della creazione. Ogni AWS regione ha il proprio 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.

La tabella seguente elenca ogni AWS regione e il relativo prefisso ID globale del datastore.

| Nome regione/Regione                                                     | Prefix |
|--------------------------------------------------------------------------|--------|
| Stati Uniti orientali (Ohio)<br>us-east-2                                | fpkhr  |
| Stati Uniti orientali (Virginia settentrionale)<br>us-east-1             | ldgnf  |
| Regione Stati Uniti occidentali (California settentrionale)<br>us-west-1 | virxk  |
| Stati Uniti occidentali (Oregon)<br>us-west-2                            | sgaui  |
| Regione Canada (Centrale)<br>ca-central-1                                | bxodz  |
| Regione Asia Pacifico (Mumbai)<br>ap-south-1                             | erpgt  |

| Nome regione/Regione                                   | Prefix |
|--------------------------------------------------------|--------|
| Regione Asia Pacifico (Tokyo)<br>ap-northeast-1        | quwsw  |
| Regione Asia Pacifico (Seoul)<br>ap-northeast-2        | lfqnh  |
| Regione Asia Pacifico (Osaka-Locale)<br>ap-northeast-3 | nlapn  |
| Regione Asia Pacifico (Singapore)<br>ap-southeast-1    | vlqxn  |
| Regione Asia Pacifico (Sydney)<br>ap-southeast-2       | vbgxd  |
| Regione Europa (Francoforte)<br>eu-central-1           | iudkw  |
| Regione Europa (Irlanda)<br>eu-west-1                  | gxeiz  |
| Regione Europa (Londra)<br>eu-west-2                   | okuqm  |
| Regione UE (Parigi)<br>eu-west-3                       | fgjhi  |
| Regione Sud America (San Paolo)<br>sa-east-1           | juxlw  |



| Nome regione/Regione                                    | Prefix |
|---------------------------------------------------------|--------|
| Regione Cina (Pechino)<br>cn-north-1                    | emvgo  |
| Regione Cina (Ningxia)<br>cn-northwest-1                | ckbem  |
| Regione Asia Pacifico (Hong Kong)<br>ap-east-1          | knjmp  |
| AWS GovCloud (Stati Uniti occidentali)<br>us-gov-west-1 | sgwui  |

- [create-replication-group](#)— Utilizzate 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 un `--global-replication-group-id` valore, ElastiCache dedurrà i valori dal gruppo di replica primario del gruppo di replica globale per i seguenti parametri. 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
```

## Aggiornamento multimotore da Redis a Valkey per OSS ElastiCache GlobalDataStore

È possibile aggiornare un gruppo di replica OSS globale Redis esistente al motore Valkey utilizzando Console, oppure. API CLI

Se disponi di un gruppo di replica OSS globale Redis esistente, puoi eseguire l'aggiornamento a Valkey specificando il nuovo motore e la versione del motore con. `modify-global-replication-group` API

Per Linux, macOS o Unix:

```
aws elasticache modify-global-replication-group \
 --global-replication-group-id myGlobalReplGroup \
 --engine valkey \
 --apply-immediately \
 --engine-version 7.2
```

Per Windows:

```
aws elasticache modify-global-replication-group ^
 --global-replication-group-id myGlobalReplGroup ^
 --engine valkey ^
 --apply-immediately ^
 --engine-version 7.2
```

Se hai un gruppo di parametri di cache personalizzato applicato al gruppo di replica OSS globale Redis esistente che desideri aggiornare, dovrai passare anche un gruppo di parametri di cache Valkey personalizzato nella richiesta. Il gruppo di parametri personalizzati Valkey di input deve avere gli stessi valori di parametri OSS statici Redis del gruppo di parametri personalizzati Redis esistente. OSS

Per Linux, macOS o Unix:

```
aws elasticache modify-global-replication-group \
 --global-replication-group-id myGlobalReplGroup \
 --engine valkey \
 --engine-version 7.2 \
 --apply-immediately \
 --cache-parameter-group-name myParamGroup
```

Per Windows:

```
aws elasticache modify-global-replication-group ^
 --global-replication-group-id myGlobalReplGroup ^
 --engine valkey ^
 --engine-version 7.2 ^
 --apply-immediately ^
 --cache-parameter-group-name myParamGroup
```

- [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-gruppo di replica](#)

```
aws elasticache rebalance-shards-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name
```

Usa help per elencare tutti i comandi disponibili ElastiCache con Valkey o Redis. OSS

```
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

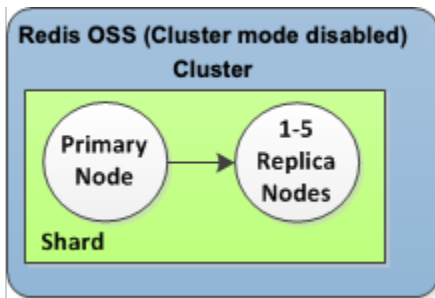
I OSS cluster Amazon ElastiCache Valkey e Redis a nodo singolo sono entità in memoria con servizi di protezione dei dati limitati (). AOF Se il cluster non viene eseguito per qualsiasi motivo, tutti i dati del cluster vengono persi. Tuttavia, se utilizzi un OSS motore Valkey o Redis, puoi raggruppare da 2 a 6 nodi in un cluster con repliche in cui da 1 a 5 nodi di sola lettura contengono dati replicati del singolo nodo primario di 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 all'interno di uno shard (chiamato gruppo di nodi in API/CLI) contenuto in un cluster Valkey o Redis. OSS I cluster Valkey o Redis OSS (modalità cluster disabilitata) hanno sempre uno shard. I cluster Valkey o Redis OSS (modalità cluster abilitata) possono avere fino a 500 shard con i dati del cluster partizionati tra gli 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 di nodi o shard può essere aumentato fino a un massimo di 500 per cluster con Valkey e con la versione 5.0.6 o successiva del motore Redis. OSS 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 insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo troppo piccolo o che le sottoreti sono condivise e utilizzate intensamente 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.



Il cluster Valkey o Redis OSS (modalità cluster disabilitata) ha uno shard 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 [Mitigazione degli errori durante l'esecuzione di Valkey o Redis OSS](#).

## Argomenti

- [Comprendere la replica di Valkey e Redis OSS](#)
- [Replica: modalità OSS cluster Valkey e Redis disabilitata o abilitata](#)
- [Riduzione al minimo dei tempi di inattività ElastiCache utilizzando Multi-AZ con Valkey e Redis OSS](#)
- [Modalità di implementazione di sincronizzazione e backup](#)
- [Creazione di un gruppo di replica Valkey o Redis OSS](#)
- [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 a quella primaria, per i gruppi di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#)

## Comprendere la replica di Valkey e Redis OSS

Redis OSS implementa la replica in due modi:

- Con un singolo shard che contiene tutti i dati del cluster in ogni nodo: Valkey o Redis (modalità cluster disabilitata) OSS
- Con dati partizionati su un massimo di 500 shard: Valkey o Redis (modalità cluster abilitata) OSS

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 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 di nodi o shard può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Redis è 5.0.6 o successiva. OSS 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 insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo troppo piccolo o che le sottoreti sono condivise e utilizzate intensamente 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.

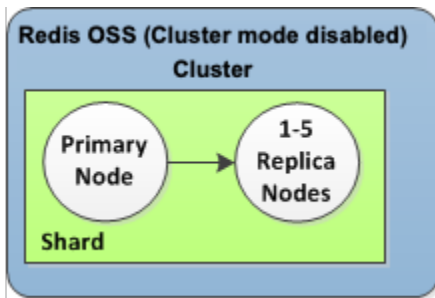
### Argomenti

- [Valkey o Redis \(modalità cluster disattivata\) OSS](#)
- [Valkey o Redis OSS \(modalità cluster abilitata\)](#)

### Valkey o Redis (modalità cluster disattivata) OSS

Un cluster Valkey o Redis OSS (modalità cluster disabilitata) ha un singolo shard, all'interno del quale è presente una raccolta di nodi: un nodo di lettura/scrittura primario e fino a cinque nodi di replica secondari di sola lettura. 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 Valkey o Redis OSS (modalità cluster disabilitata) con un singolo shard e nodi di replica

È possibile utilizzare i cluster Valkey o Redis OSS (modalità cluster disabilitata) con nodi di replica per scalare la soluzione in modo da gestire applicazioni che richiedono un uso intensivo di lettura o ElastiCache per supportare un gran numero di client che leggono contemporaneamente dallo stesso cluster.

Tutti i nodi di un cluster Valkey o Redis OSS (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 agli errori e ridurre i tempi di inattività in scrittura, abilita Multi-AZ con failover automatico per il cluster Valkey o Redis OSS (modalità cluster disabilitata) con repliche. Per ulteriori informazioni, consulta [Riduzione al minimo dei tempi di inattività ElastiCache utilizzando Multi-AZ con Valkey e Redis OSS](#).

È possibile modificare i ruoli dei nodi all'interno del cluster Valkey o Redis OSS (modalità cluster disabilitata), scambiando i ruoli tra la replica principale e quella delle repliche. 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 a quella primaria, per i gruppi di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

Valkey o Redis OSS (modalità cluster abilitata)

Un cluster Valkey o Redis OSS (modalità cluster abilitata) è composto da 1 a 500 shard (API/ gruppi di nodi). CLI 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.

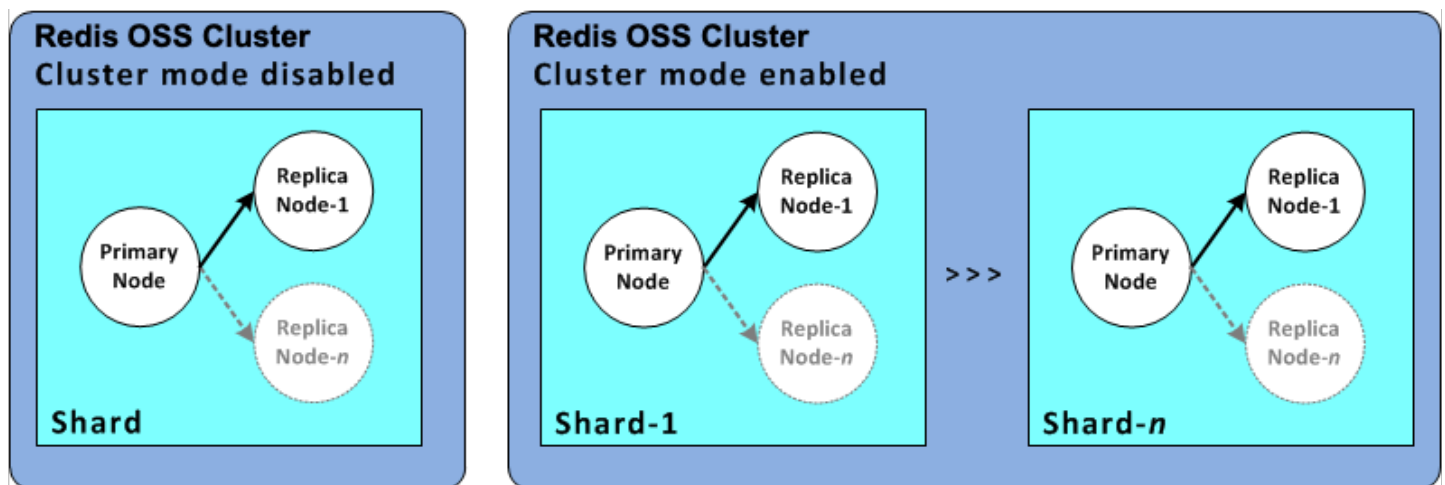
Il limite di nodi o shard può essere aumentato fino a un massimo di 500 per cluster se la versione del motore è Valkey 7.2 e superiore o Redis 5.0.6 e successiva. OSS Ad esempio, è possibile scegliere di configurare un cluster a 500 nodi che varia tra 83 partizioni (un primario e 5 repliche per partizione) e 500 partizioni (un singolo primario e nessuna replica). Assicurati che esistano abbastanza indirizzi IP disponibili per soddisfare l'aumento. Le insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo troppo piccolo o che le sottoreti sono condivise e utilizzate intensamente 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 Valkey o Redis (modalità cluster abilitata). OSS

Le applicazioni utilizzano l'endpoint di configurazione del cluster Valkey o Redis OSS (modalità cluster abilitata) per connettersi con i nodi del cluster. Per ulteriori informazioni, consulta [Ricerca degli endpoint di connessione in ElastiCache](#).



Cluster Valkey o Redis OSS (modalità cluster abilitata) con più shard e nodi di replica

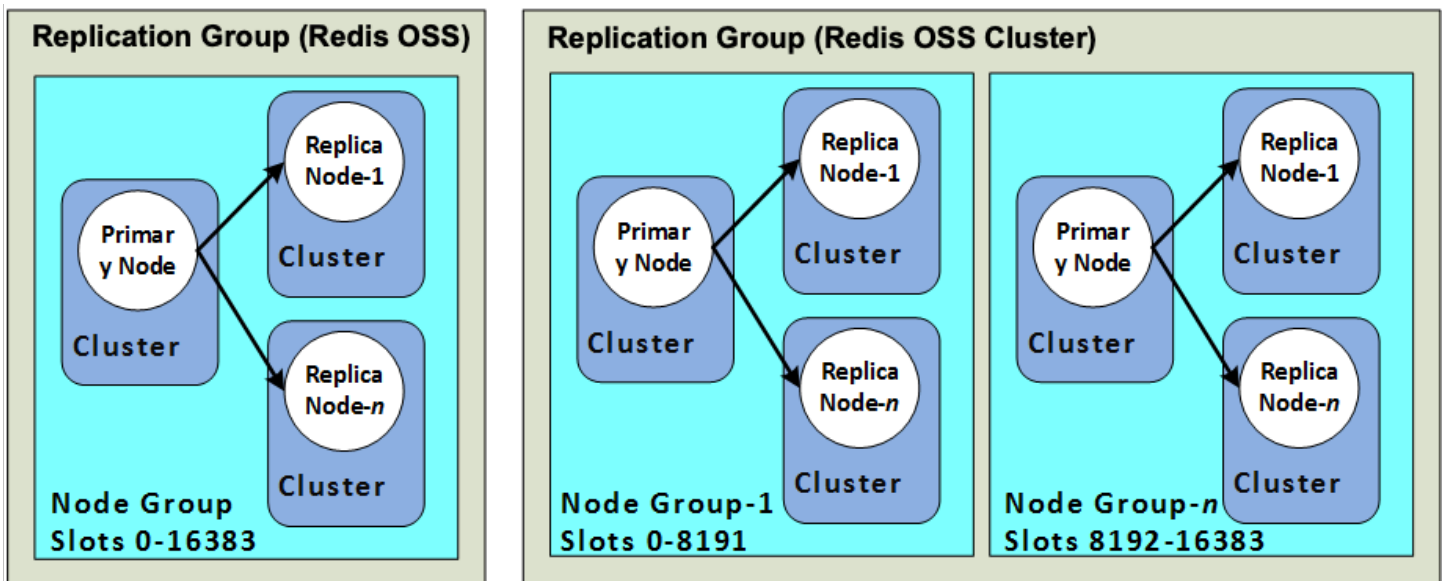
Tutti i nodi di un cluster Valkey o Redis OSS (modalità cluster abilitata) devono risiedere nella stessa regione. Per migliorare la tolleranza ai guasti, puoi eseguire il provisioning dei nodi primari e delle repliche di lettura in più zone di disponibilità all'interno di tale regione.

Attualmente, le funzionalità di Valkey o Redis OSS (modalità cluster abilitata) presentano alcune limitazioni.

- Non puoi promuovere manualmente nodi di replica al nodo primario.

## Replica: modalità OSS cluster Valkey e Redis disabilitata o abilitata



A partire da Valkey 7.2 e Redis OSS versione 3.2, è possibile creare uno dei due tipi distinti di cluster (/: gruppi di replica). API CLI Un cluster Valkey o Redis OSS (modalità cluster disabilitata) ha sempre un singolo shard (API/CLI: gruppo di nodi) con un massimo di 5 nodi di replica in lettura. Un cluster Valkey o Redis OSS (modalità cluster abilitata) ha fino a 500 shard con da 1 a 5 nodi di replica di lettura ciascuno.



Cluster Valkey o Redis OSS (modalità cluster disabilitata) e Valkey o Redis (modalità cluster abilitata) OSS

La tabella seguente riassume le differenze importanti tra i cluster Valkey o Redis OSS (modalità cluster disabilitata) e i cluster Valkey o Redis (modalità cluster abilitata). OSS

## Confronto tra i cluster Valkey o Redis OSS (modalità cluster disabilitata) e Valkey o Redis (modalità cluster abilitata) OSS

| Funzionalità             | Valkey o Redis (modalità cluster disabilitata) OSS                                                                                                                                                        | Valkey o Redis OSS (modalità cluster abilitata)                                                                                                                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Modificabile             | Sì. Supporta l'aggiunta e l'eliminazione di nodi di replica e il dimensionamento orizzontale del tipo di nodo.                                                                                            | Limitato. Per ulteriori informazioni, consulta <a href="#">Gestione delle versioni per ElastiCache</a> e <a href="#">Scalabilità dei cluster in Valkey o Redis OSS (modalità cluster abilitata)</a> .                                     |
| Partizionamento dei dati | No                                                                                                                                                                                                        | Sì                                                                                                                                                                                                                                        |
| Shard                    | 1                                                                                                                                                                                                         | Da 1 a 500                                                                                                                                                                                                                                |
| Repliche di lettura      | Da 0 a 5<br><br> <b>Important</b><br>Se non esistono repliche e il nodo non viene eseguito, tutti i dati vengono persi. | Da 0 a 5 per partizione.<br><br> <b>Important</b><br>Se non esistono repliche e il nodo non viene eseguito, tutti i dati in tale shard vengono persi. |
| Multi-AZ                 | Sì, con almeno 1 replica.<br><br>Facoltativo. Attivata per impostazione predefinita.                                                                                                                      | Sì<br><br>Facoltativo. Attivata per impostazione predefinita.                                                                                                                                                                             |
| Snapshot (Backup)        | Sì, creando un singolo file .rdb.                                                                                                                                                                         | Sì, creando un file .rdb univoco per ogni shard.                                                                                                                                                                                          |
| Ripristino               | Sì, utilizzando un singolo file.rdb da un cluster Valkey o Redis OSS (modalità cluster disabilitata).                                                                                                     | Sì, utilizzando file.rdb da un cluster Valkey o Redis OSS (modalità cluster disabilitata) o da un cluster Valkey o Redis                                                                                                                  |

|                     |                                                                                                                                                           |                                                                                                                                            |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Funzionalità        | Valkey o Redis (modalità cluster disabilitata) OSS                                                                                                        | Valkey o Redis OSS (modalità cluster abilitata)<br><br>(modalità cluster abilitata).<br>OSS                                                |
| Supportato da       | Tutte OSS le versioni di Valkey e Redis                                                                                                                   | Tutte le versioni di Valkey e Redis OSS 3.2 e successive                                                                                   |
| Motore aggiornabile | Sì, con alcuni limiti. Per ulteriori informazioni, consulta <a href="#">Gestione delle versioni per ElastiCache</a> .                                     | Sì, con alcuni limiti. Per ulteriori informazioni, consulta <a href="#">Gestione delle versioni per ElastiCache</a> .                      |
| Crittografia        | Versioni 3.2.6 (prevista perEOL, consulta la <a href="#">pianificazione di fine del ciclo di vita delle OSS versioni Redis</a> ) e 4.0.10 e successive.   | Versioni 3.2.6 (prevista perEOL, consulta la <a href="#">pianificazione di fine vita delle OSS versioni Redis</a> ) e 4.0.10 e successive. |
| HIPAAldoneo         | Versioni 3.2.6 (pianificata perEOL, vedi la <a href="#">pianificazione della fine del ciclo di vita delle OSS versioni Redis</a> ) e 4.0.10 e successive. | Versioni 3.2.6 (prevista perEOL, consulta la <a href="#">pianificazione di fine vita delle OSS versioni Redis</a> ) e 4.0.10 e successive. |
| PCIDSSConforme      | Versioni 3.2.6 (prevista perEOL, vedi la <a href="#">pianificazione della fine del ciclo di vita delle OSS versioni Redis</a> ) e 4.0.10 e successive.    | Versioni 3.2.6 (prevista perEOL, consulta la <a href="#">pianificazione di fine vita delle OSS versioni Redis</a> ) e 4.0.10 e successive. |
| Resharding online   | N/D                                                                                                                                                       | Versione 3.2.10 (prevista perEOL, vedi la pianificazione di <a href="#">fine vita delle OSS versioni Redis</a> ) e successive.             |

scegliere il servizio più adatto.

Quando scegli tra Valkey o Redis OSS (modalità cluster disabilitata) o Valkey o Redis OSS (modalità cluster abilitata), considera i seguenti fattori:

- Dimensionamento e partizionamento - Le esigenze aziendali cambiano. Occorre fornire il provisioning per picchi di richiesta o ridimensionare quando la richiesta cambia. Valkey o Redis OSS (modalità cluster disabilitata) supportano la scalabilità. 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, consultare [Scalabilità dei nodi di replica per Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

Valkey o Redis OSS (modalità cluster abilitata) supportano il partizionamento dei dati su 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 ridimensionamento delle partizioni, consulta [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)

- Dimensione del nodo rispetto al numero di nodi: poiché un cluster Valkey o Redis OSS (modalità cluster disabilitata) ha un solo shard, il tipo di nodo deve essere sufficientemente grande da contenere tutti i dati del cluster più il sovraccarico necessario. D'altra parte, poiché è possibile partizionare i dati su più shard quando si utilizza un cluster Valkey o Redis OSS (modalità cluster abilitata), i tipi di nodi possono essere più piccoli, ma è necessario un numero maggiore di nodi. Per ulteriori informazioni, consulta [Scelta delle dimensioni dei nodi](#).
- Letture o scritture: se il carico principale sul cluster è costituito da applicazioni che leggono dati, è possibile scalare un cluster Valkey o Redis OSS (modalità cluster disattivata) aggiungendo ed eliminando repliche di lettura. Tuttavia, esiste un numero massimo di 5 repliche di lettura. Se il carico sul cluster è elevato in termini di scrittura, puoi trarre vantaggio dagli endpoint di scrittura aggiuntivi di un cluster Valkey o Redis OSS (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 al minimo dei tempi di inattività ElastiCache utilizzando Multi-AZ con Valkey e Redis OSS

Esistono diversi casi in cui ElastiCache con Valkey e Redis OSS potrebbe essere necessario sostituire un nodo primario, tra cui alcuni tipi di manutenzione pianificata e l'improbabile eventualità di un guasto del nodo primario o della 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 ed effettuare il provisioning di un nuovo nodo primario, in quanto ElastiCache gestirà la situazione 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 anche il nome Domain Name Service (DNS) 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 i OSS cluster ElastiCache Valkey e Redis, le sostituzioni pianificate dei nodi vengono completate mentre il cluster soddisfa le richieste di scrittura in entrata.
- Per i cluster Valkey e Redis disabilitati in modalità OSS cluster con Multi-AZ abilitato che funzionano sul motore 5.0.6 o successivo, le sostituzioni pianificate dei nodi vengono completate mentre il cluster gestisce le richieste di scrittura in entrata.
- Per i cluster Valkey e Redis disabilitati in modalità OSS cluster con Multi-AZ abilitato che funzionano sul motore 4.0.10 o precedente, potresti notare 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.

È possibile abilitare Multi-AZ utilizzando la console di gestione, o il ElastiCache . AWS CLI ElastiCache API

L'abilitazione di ElastiCache Multi-AZ sul OSS cluster Valkey o Redis (nel gruppo di replica API e CLI nel gruppo di replica) migliora la tolleranza agli errori. Questo vale in particolare nei casi in cui il cluster principale di lettura/scrittura del cluster diventa irraggiungibile o si guasta per qualche motivo. Multi-AZ è supportato solo sui OSS cluster Valkey e Redis con più di un nodo in ogni shard.

## Argomenti

- [Abilitazione della funzione Multi-AZ](#)
- [Risposte per scenari di errore relativi alla funzione Multi-AZ](#)
- [Test del failover automatico](#)
- [Limitazioni su Multi-AZ](#)

## Abilitazione della funzione Multi-AZ

È possibile abilitare Multi-AZ quando si crea o si modifica un cluster (API o CLI un gruppo di replica) utilizzando la console o il. ElastiCache AWS CLI ElastiCache API

È possibile abilitare Multi-AZ solo su cluster Valkey o Redis OSS (modalità cluster disabilitata) che dispongono di almeno una replica di lettura disponibile. 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 Valkey o Redis OSS](#). Per informazioni sull'aggiunta di una replica di lettura a un cluster con replica, consulta [Aggiungere una replica di lettura per Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

## Argomenti

- [Abilitazione del Multi-AZ \(console\)](#)
- [Abilitazione di Multi-AZ \(AWS CLI\)](#)
- [Abilitazione di Multi-AZ \(ElastiCache API\)](#)

## Abilitazione del Multi-AZ (console)

È possibile abilitare Multi-AZ utilizzando la ElastiCache console quando si crea un nuovo cluster Valkey o Redis o modificando un OSS cluster esistente con la replica.

Multi-AZ è abilitato per impostazione predefinita sui cluster Valkey o OSS Redis (modalità cluster abilitata).



**⚠ Important**

ElastiCache abiliterà automaticamente Multi-AZ solo se il cluster contiene almeno una replica in una zona di disponibilità diversa da quella principale in tutti gli shard.

Abilitazione di Multi-AZ durante la creazione di un cluster utilizzando la console ElastiCache

Per ulteriori informazioni su questo processo, consulta [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#). 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 [Usando il ElastiCache AWS Management Console](#) nella sezione relativa alla modifica di un cluster.

Abilitazione di Multi-AZ (AWS CLI)

Il seguente esempio di codice utilizza AWS CLI per abilitare 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'JSONOutput di questo comando dovrebbe essere simile al seguente.

```
{
 "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 AWS CLI Command Reference.

### Abilitazione di Multi-AZ (ElastiCache API)

Il seguente esempio di codice utilizza ElastiCache API per abilitare 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 Guida di riferimento: ElastiCache API

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

## Risposte per scenari di errore relativi alla funzione Multi-AZ

Prima dell'introduzione di Multi-AZ, ElastiCache rilevava e sostituiva i nodi guasti di un cluster ricreando e rifornendo il nodo guasto. 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 Multi-AZ è abilitato, monitora ElastiCache continuamente lo stato del nodo primario. Se il nodo primario non riesce, viene eseguita una delle seguenti operazioni a seconda del tipo di errore.

### Argomenti

- [Scenari di errore quando solo il nodo primario non riesce](#)
- [Scenari di errore 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 si verifica un errore solo nel nodo primario, 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 sull'endpoint primario, non è necessario modificare l'endpoint per operazioni di scrittura o lettura. ElastiCachepropaga il DNS nome 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 all'applicazione, poiché il DNS nome 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 Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint per i gruppi di OSS replica Valkey o Redis \(\)AWS CLI](#)
- [Ricerca degli endpoint per i gruppi di replica Valkey o Redis \(\) OSS ElastiCache API](#)

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 falliscono, 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 sull'endpoint primario, non è necessario modificare l'endpoint per le scritture. ElastiCache propaga il DNS nome 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 DNS nome 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 Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint per i gruppi di OSS replica Valkey o Redis \(\)AWS CLI](#)
- [Ricerca degli endpoint per i gruppi di replica Valkey o Redis \(\) OSS ElastiCache API](#)

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.

In caso di guasto dell'intero cluster, 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 Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)
- [Individuazione degli endpoint per i gruppi di OSS replica Valkey o Redis \(\)AWS CLI](#)
- [Ricerca degli endpoint per i gruppi di replica Valkey o Redis \(\) OSS ElastiCache API](#)

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 il failover automatico, è possibile testarlo utilizzando la ElastiCache console AWS CLI, il e il. ElastiCache API

Durante il test, tieni presente quanto segue:

- È possibile utilizzare questa operazione per testare il failover automatico su un massimo di 15 shard (chiamati gruppi di nodi in the ElastiCache API and AWS CLI) in un periodo di 24 ore consecutive.
- Se si chiama questa operazione su shard in cluster diversi (denominati gruppi di replica in API and CLI), è possibile effettuare le chiamate contemporaneamente.
- In alcuni casi, è possibile richiamare questa operazione più volte su shard diversi nello stesso gruppo di replica Valkey o Redis OSS (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 completa, controlla gli eventi utilizzando la ElastiCache console Amazon, il AWS CLI, o il ElastiCache API. 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 ElastiCache degli eventi](#) nella Guida per l'utente di ElastiCache
- [DescribeEvents](#) nel ElastiCache API Reference
- [describe-events](#) nel Riferimento ai comandi AWS CLI .
- API È progettato per testare il comportamento dell'applicazione in caso di ElastiCache failover. 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 bloccarlo. API

## Argomenti

- [Test del failover automatico utilizzando AWS Management Console](#)
- [Test del failover automatico utilizzando AWS CLI](#)
- [Test del failover automatico utilizzando ElastiCache API](#)

### Test del failover automatico utilizzando AWS Management Console

Utilizza la procedura seguente per testare il failover automatico con la console.

Per testare il failover automatico

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nel pannello di navigazione, scegli Valkey o Redis OSS.
3. Dall'elenco dei cluster, 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 [Usando il ElastiCache AWS Management Console](#).



5. Per Valkey o Redis OSS (modalità cluster disabilitata), scegli il nome del cluster.

Per Valkey o Redis OSS (modalità cluster abilitata), procedi come segue:

- a. Seleziona il nome del cluster.

- b. Nella pagina Shards, per lo shard (chiamato gruppo di nodi in API and CLI) su cui desiderate testare il failover, scegliete il nome dello shard.
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 AWS CLI

È possibile testare il failover automatico su qualsiasi cluster dotato di Multi-AZ utilizzando l' AWS CLI operazione. `test-failover`

### 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.
- È possibile testare un massimo di 15 gruppi di nodi in un periodo continuativo di 24 ore.

L'esempio seguente lo utilizza AWS CLI per testare il failover automatico sul gruppo di nodi `redis00-0003` nel cluster Valkey o Redis OSS (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 tenere traccia dell'avanzamento del failover, utilizza l'operazione. AWS CLI `describe-events`

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 utilizzando ElastiCache API

È possibile testare il failover automatico su qualsiasi cluster abilitato con Multi-AZ utilizzando l'ElastiCache API operazione. `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. È possibile testare un massimo di 15 gruppi di nodi in un periodo continuativo 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 tenere traccia dell'avanzamento del failover, utilizzate l' `ElastiCache DescribeEventsAPI` operazione.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- [TestFailover](#) nel riferimento ElastiCache API
- [DescribeEvents](#) nel riferimento ElastiCache API

## Limitazioni su Multi-AZ


Tieni presente le seguenti limitazioni per Multi-AZ:

- Multi-AZ è supportato su Valkey e sulla OSS versione Redis 2.8.6 e successive.
- Multi-AZ non è supportato sui tipi di nodi T1.
- La replica di Valkey e Redis è OSS 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 principale, ElastiCache sceglie la replica con il minor ritardo di replica. 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 le repliche di lettura a primarie su cluster Valkey o Redis con la modalità OSS cluster disabilitata, puoi farlo solo quando Multi-AZ e il failover automatico sono disabilitati. 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. È possibile eseguire questa operazione tramite la console deselezionando la casella di controllo Auto failover per il gruppo di replica. È inoltre possibile eseguire questa operazione AWS CLI impostando la `AutomaticFailoverEnabled` proprietà su `false` quando si chiama l'operazione. `ModifyReplicationGroup`
  3. Promuovere la replica di lettura al nodo primario.
  4. Riattivare la funzione Multi-AZ.
- ElastiCache (RedisOSS) Multi-AZ e append-only file (AOF) si escludono a vicenda. 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 (dove si trovava il file primario guasto) solo quando Az-a è di nuovo attivo e 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 nativo di Valkey e Redis e non è OSS

esclusivo di Multi-AZ. ElastiCache Per i dettagli su questo comportamento, consulta [Replication](#) sul sito Web di Valkey.

 Important

Per Valkey 7.2.6 e versioni successive o Redis OSS versione 2.8.22 e successive, non è possibile creare repliche esterne.

Per OSS le versioni Redis precedenti alla 2.8.22, si consiglia di non connettere una replica esterna a un cluster con funzionalità Multi-AZ. ElastiCache Questa configurazione non supportata può creare problemi che impediscono la corretta esecuzione del failover e del ElastiCache ripristino. Per connettere una replica esterna a un ElastiCache cluster, assicurati che Multi-AZ non sia abilitato prima di effettuare la connessione.



## Modalità di implementazione di sincronizzazione e backup

Tutte le versioni supportate di Valkey e Redis OSS supportano il backup e la sincronizzazione tra i nodi primari e di replica. Tuttavia, il modo in cui vengono implementati il backup e la sincronizzazione varia a seconda della versione.

### Redis OSS versione 2.8.22 e successive

La OSS replica Redis, nelle versioni 2.8.22 e successive, consente di scegliere tra due metodi. Per ulteriori informazioni, consulta [Versioni Redis precedenti alla 2.8.22 OSS](#) 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 alla 2.8.22 OSS

Il OSS backup e la sincronizzazione di Redis nelle versioni precedenti alla 2.8.22 sono un processo in tre fasi.

1. Fork e il processo in background serializzano i dati del cluster su disco. Questo crea un'istantanea. 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 [Assicurarsi di disporre di memoria sufficiente per creare un'istantanea Valkey o Redis OSS](#).

3. Infine, trasmettere i dati di cache e il change log al nodo di replica.

## Creazione di un gruppo di replica Valkey o Redis OSS

Per creare un cluster con nodi di replica sono disponibili le seguenti opzioni. Uno si applica quando è già disponibile un cluster Valkey o Redis OSS (modalità cluster disabilitata) non associato a nessun cluster con repliche da utilizzare come nodo principale. L'altra si applica quando desideri creare un nodo primario con il cluster e le repliche di lettura. Attualmente, un cluster Valkey o Redis OSS (modalità cluster abilitata) deve essere creato da zero.

### Opzione 1: [Creazione di un gruppo di replica utilizzando un cluster esistente](#)

Utilizza questa opzione per sfruttare un cluster Valkey o Redis a nodo singolo esistente OSS (modalità cluster disabilitata). 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, consulta [Creazione di un gruppo di replica utilizzando un cluster esistente](#).

#### Important

Non è possibile creare un cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando un cluster esistente. Per creare un cluster Valkey o Redis OSS (modalità cluster abilitata) (API/CLI: gruppo di replica) utilizzando la console, vedere. ElastiCache [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)

### Opzione 2: [Creazione da zero di un gruppo di OSS replica Valkey o Redis](#)

Utilizza questa opzione se non disponi già di un cluster Valkey o Redis OSS (modalità cluster disabilitata) da utilizzare come nodo primario del cluster o se desideri creare un cluster Valkey o Redis OSS (modalità cluster abilitata). Per informazioni, consulta [Creazione da zero di un gruppo di OSS replica Valkey o Redis](#).

## Creazione di un gruppo di replica utilizzando un cluster esistente

Un cluster disponibile è un cluster Valkey o OSS Redis a nodo singolo esistente. Attualmente, Valkey o Redis OSS (modalità cluster abilitata) non supportano la creazione di un cluster con repliche utilizzando un cluster a nodo singolo disponibile. Se desideri creare un cluster Valkey o Redis OSS (modalità cluster abilitata), consulta. [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)

La procedura seguente può essere utilizzata solo se si dispone di un cluster a nodo singolo Valkey o Redis OSS (modalità cluster disabilitata). Il nodo di questo cluster diventa il nodo primario nel nuovo cluster. Se non disponi di un cluster Valkey o Redis OSS (modalità cluster disabilitata) che puoi utilizzare come primario del nuovo cluster, vedi. [Creazione da zero di un gruppo di OSS replica Valkey o Redis](#)

Creazione di un gruppo di replica utilizzando un cluster esistente (Console)

Consulta l'argomento [Utilizzando il ElastiCache AWS Management Console](#)

Creazione di un gruppo di replica utilizzando un cluster di cache Valkey o Redis OSS disponibile  
()AWS CLI

Esistono due passaggi per creare un gruppo di replica con repliche di lettura quando si utilizza un Valkey o Redis OSS Cache Cluster disponibile come primario quando si utilizza il. AWS CLI

Quando si utilizza il, AWS CLI si crea un gruppo di replica specificando il nodo autonomo disponibile come nodo principale del cluster `--primary-cluster-id` e il numero di nodi che si desidera inserire 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 Valkey o Redis OSS (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 Valkey o Redis OSS (modalità cluster disabilitata) disponibile che si desidera utilizzare come nodo principale in questo gruppo di replica.

Il comando seguente crea il gruppo di replica `sample-repl-group` utilizzando il cluster Valkey o Redis OSS (modalità cluster disabilitata) disponibile `redis01` 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 ulteriori informazioni e parametri che potresti voler utilizzare, consulta l'argomento [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 \
 --primary-cluster-id sample-cluster \
 --num-read-replicas 1
```

```
--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 AWS CLI argomenti:

- [create-replication-group](#)
- [modify-replication-group](#)

Aggiungere repliche a un cluster Valkey o Redis OSS (Cluster Mode Disabled) autonomo ()  
ElastiCache API

Quando si utilizza il ElastiCache API, si crea un gruppo di replica specificando il nodo autonomo disponibile come nodo principale del cluster `PrimaryClusterId` e il numero di nodi che si desidera

inserire 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 Valkey o Redis OSS (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 Valkey o Redis disponibile OSS (modalità cluster disabilitata) che si desidera utilizzare come nodo principale di questo cluster.

Il comando seguente crea il cluster con repliche `sample-repl-group` utilizzando il cluster Valkey o Redis OSS (modalità cluster disabilitata) disponibile `redis01` 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.

```
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 ElastiCache APL:

- [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=myReplGroup
&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 ulteriori informazioni e parametri che potresti voler utilizzare, consulta l'argomento [ElastiCache API `CreateCacheCluster`](#).

## Creazione da zero di un gruppo di OSS replica Valkey o Redis

Di seguito, è possibile scoprire come creare un gruppo di OSS replica Valkey o Redis senza utilizzare un cluster Valkey o Redis esistente come principale. OSS È possibile creare da zero un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) o Valkey o Redis OSS (modalità cluster abilitata) utilizzando la console, il o il. ElastiCache AWS CLI ElastiCache API

Prima di continuare, decidete se volete creare un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) o Valkey o OSS Redis (modalità cluster abilitata). Per le linee guida, consulta [Replica: modalità OSS cluster Valkey e Redis disabilitata o abilitata](#).

### Argomenti

- [Creazione da zero di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#)
- [Creazione da zero di un gruppo di replica in Valkey o Redis OSS \(modalità cluster abilitata\)](#)

## Creazione da zero di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata)

È possibile creare un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) partendo da zero utilizzando la ElastiCache console, il, o il. AWS CLI ElastiCache API Un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) ha sempre un gruppo di nodi, un cluster primario e fino a cinque repliche di lettura. I gruppi di replica Valkey o Redis OSS (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 da zero un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata), utilizzate uno dei seguenti approcci:

Creazione da zero di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) (AWS CLI)

La procedura seguente crea un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) utilizzando. AWS CLI

Quando si crea un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) partendo da zero, si crea il gruppo di replica e tutti i relativi nodi con una singola chiamata al comando. 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 Valkey o Redis OSS (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 delle prestazioni per ogni tipo di nodo, consulta [Amazon EC2 Instance Types](#).

**--data-tiering-enabled**

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 [Suddivisione dei dati su più livelli in ElastiCache](#).

**--cache-parameter-group**

Specifica un gruppo di parametri che corrisponde alla versione motore. Se utilizzi Redis OSS 3.2.4 o versione successiva, specifica il gruppo di parametri o un gruppo di `default.redis3.2` parametri derivato da `default.redis3.2` cui creare un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Parametri Valkey e Redis OSS](#).

**--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 la versione 3.2.6 o 4.0.10 di Redis. OSS
- Il gruppo di replica deve essere creato in AmazonVPC.
- Devi anche includere il parametro `--cache-subnet-group`.
- È inoltre necessario includere il parametro `--auth-token` con il valore di stringa specificato dal cliente per il AUTH token (password) necessario per eseguire operazioni su questo gruppo di replica.

La seguente operazione crea un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) `sample-repl-group` con tre nodi, uno principale e due repliche.

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 ulteriori informazioni e parametri che potresti voler utilizzare, consulta l'argomento. [AWS CLI create-replication-group](#)

Creazione da zero di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) ()  
ElastiCache API

La procedura seguente crea un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) utilizzando. ElastiCache API

Quando si crea un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) partendo da zero, si crea il gruppo di replica e tutti i relativi nodi con una singola chiamata all'operazione. ElastiCache API `CreateReplicationGroup` Include i seguenti parametri.

### ReplicationGroupId

Il nome del gruppo di replica che stai creando.

I vincoli di denominazione dei gruppi di replica Valkey o Redis OSS (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 delle prestazioni per ogni tipo di nodo, consulta [Amazon EC2 Instance Types](#).

## --data-tiering-enabled

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 [Suddivisione dei dati su più livelli in ElastiCache](#).

## CacheParameterGroup

Specifica un gruppo di parametri che corrisponde alla versione motore. Se utilizzi Redis OSS 3.2.4 o versione successiva, specifica il gruppo di parametri o un gruppo di `default.redis3.2` parametri derivato da `default.redis3.2` cui creare un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata). Per ulteriori informazioni, consulta [Parametri Valkey e Redis OSS](#).

## --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 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 TransitEncryptionEnabled=true o AtRestEncryptionEnabled=true e soddisfa le seguenti condizioni.

- Il gruppo di replica deve eseguire la versione 3.2.6 o 4.0.10 di Redis. OSS
- Il gruppo di replica deve essere creato in AmazonVPC.
- Devi anche includere il parametro CacheSubnetGroup.
- È inoltre necessario includere il parametro AuthToken con il valore di stringa specificato dal cliente per il AUTH token (password) necessario per eseguire operazioni su questo gruppo di replica.

La seguente operazione crea il gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) myReplGroup con tre nodi, uno principale e due repliche.

```
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=myReplGroup
 &Version=2015-02-02
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20150202T192317Z
 &X-Amz-Credential=<credential>
```



Per ulteriori informazioni e parametri che potresti voler utilizzare, consulta l'argomento ElastiCache API [CreateReplicationGroup](#).

## Creazione da zero di un gruppo di replica in Valkey o Redis OSS (modalità cluster abilitata)

È possibile creare un cluster Valkey o Redis OSS (modalità cluster abilitata) (API/CLI: gruppo di replica) utilizzando la ElastiCache console, il, o il. AWS CLI ElastiCache API Un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) ha da 1 a 500 shard (API/CLI: gruppi di nodi), un nodo primario in ogni shard e fino a 5 repliche di lettura in ogni 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 di nodi o shard può essere aumentato fino a un massimo di 500 per cluster se la versione del motore Valkey o Redis è 5.0.6 o successiva. OSS 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 insidie più comuni includono che le sottoreti del gruppo di sottoreti hanno un CIDR intervallo troppo piccolo o che le sottoreti sono condivise e utilizzate intensamente 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.

### Creazione di un cluster in Valkey o Redis (modalità cluster abilitata) OSS

- [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)
- [Creazione da zero di un gruppo di replica Valkey o Redis OSS \(Cluster Mode Enabled\) \(AWS CLI\)](#)
- [Creazione da zero di un gruppo di replica in Valkey o Redis OSS \(modalità cluster abilitata\) \(ElastiCache API\)](#)

### Creazione di un cluster Valkey o Redis OSS (modalità cluster abilitata) (console)

Per creare un cluster Valkey o Redis OSS (modalità cluster abilitata), vedere. [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#) 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 Valkey o Redis OSS (Cluster Mode Enabled) ()AWS CLI

La procedura seguente crea un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) utilizzando AWS CLI

Quando si crea un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) partendo da zero, si crea il gruppo di replica e tutti i relativi nodi con una singola chiamata al comando 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 Valkey o Redis OSS (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 delle prestazioni per ogni tipo di nodo, consulta [Amazon EC2 Instance Types](#).

`--data-tiering-enabled`

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 [Suddivisione dei dati su più livelli in ElastiCache](#).

## --cache-parameter-group

Specificate il gruppo di `default.redis6.x.cluster.on` parametri o un gruppo di parametri derivato da `default.redis6.x.cluster.on` cui creare un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata). Per ulteriori informazioni, consulta [Modifiche ai parametri di Redis OSS 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, ipv 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 OSS versione 3.2.6 o 4.0.10.
- Il gruppo di replica deve essere creato in AmazonVPC.

- Devi anche includere il parametro `--cache-subnet-group`.
- È inoltre necessario includere il parametro `--auth-token` con il valore di stringa specificato dal cliente per il AUTH token (password) necessario per eseguire operazioni su questo gruppo di replica.

La seguente operazione crea il gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) `sample-repl-group` con tre gruppi/shard di nodi (`--num-node-groups`), ciascuno con tre nodi, una replica primaria e due repliche di lettura (`--replicas-per-node-group`). `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 si crea un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) da zero, è possibile configurare ogni shard del cluster utilizzando il `--node-group-configuration` parametro come mostrato nell'esempio seguente che configura due gruppi di nodi (Console: shards). 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 questo parametro viene omissso, ElastiCache sceglie la zona di disponibilità per il 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 questo parametro viene omissso, 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 questo parametro viene omissso, ElastiCache alloca le chiavi in modo equo tra i gruppi di nodi.

Esempio: "0-4999"

La seguente operazione crea il gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) `new-group` con due gruppi di nodi/shard (). `--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 ulteriori informazioni e parametri che potresti voler utilizzare, consulta l'argomento [AWS CLI create-replication-group](#).

Creazione da zero di un gruppo di replica in Valkey o Redis OSS (modalità cluster abilitata) ()  
ElastiCache API

La procedura seguente crea un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) utilizzando. ElastiCache API

Quando si crea un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) partendo da zero, si crea il gruppo di replica e tutti i relativi nodi con una singola chiamata all'operazione. ElastiCache API `CreateReplicationGroup` Include i seguenti parametri.

**ReplicationGroupId**

Il nome del gruppo di replica che stai creando.

I vincoli di denominazione dei gruppi di replica Valkey o Redis OSS (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 questo parametro viene omissso, ElastiCache sceglie la zona di disponibilità per il 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 questo parametro viene omissso, 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 questo parametro viene omissso, ElastiCache alloca le chiavi in modo equo 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 delle prestazioni per ogni tipo di nodo, consulta [Amazon EC2 Instance Types](#).

### --data-tiering-enabled

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 [Suddivisione dei dati su più livelli in ElastiCache](#).

## CacheParameterGroup

Specificate il gruppo di default.redis6.x.cluster.on parametri o un gruppo di parametri derivato da default.redis6.x.cluster.on cui creare un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata). Per ulteriori informazioni, consulta [Modifiche ai parametri di Redis OSS 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 OSS versione 3.2.6 o 4.0.10.
- Il gruppo di replica deve essere creato in AmazonVPC.

- Devi anche includere il parametro `CacheSubnetGroup`.
- È inoltre necessario includere il parametro `AuthToken` con il valore di stringa specificato dal cliente per il AUTH token (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 ulteriori informazioni e parametri che potreste voler utilizzare, consultate l'argomento [ElastiCache API `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 ElastiCache console, il AWS CLI for ElastiCache o il ElastiCache API. Il processo della console è diverso per Valkey o Redis OSS (modalità cluster disabilitata) e Valkey o Redis OSS (modalità cluster abilitata).

### Visualizzazione dei dettagli del gruppo di replica

- [Visualizzazione di un Valkey o Redis OSS \(modalità cluster disabilitata\) con repliche](#)
  - [Visualizzazione di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)
  - [Visualizzazione di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\) \(AWS CLI\)](#)

- [Visualizzazione di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\) \(\) ElastiCache API](#)
- [Visualizzazione di un gruppo di replica: Valkey o Redis OSS \(modalità cluster abilitata\)](#)
  - [Visualizzazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)
  - [Visualizzazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(\)AWS CLI](#)
  - [Visualizzazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(\) ElastiCache API](#)
- [Visualizzazione dei dettagli di un gruppo di replica \(AWS CLI\)](#)
- [Visualizzazione dei dettagli di un gruppo di replica \(ElastiCache API\)](#)

Visualizzazione di un Valkey o Redis OSS (modalità cluster disabilitata) con repliche

È possibile visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster disabilitata) con repliche (API/CLI: gruppo di replica) utilizzando la ElastiCache console, il for o il. AWS CLI ElastiCache ElastiCache API

Visualizzazione dei dettagli di un cluster Valkey o Redis OSS (modalità cluster disabilitata)

- [Visualizzazione di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)
- [Visualizzazione di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\) \(\)AWS CLI](#)
- [Visualizzazione di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\) \(\) ElastiCache API](#)

Visualizzazione di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) (console)

Per visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster disabilitata) con repliche utilizzando la ElastiCache console, consulta l'argomento. [Visualizzazione dei dettagli di Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)

Visualizzazione di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) ()AWS CLI

Per un AWS CLI esempio che mostra i dettagli di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata), vedere. [Visualizzazione dei dettagli di un gruppo di replica \(AWS CLI\)](#)

## Visualizzazione di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) () ElastiCache API

Per un ElastiCache API esempio che mostra i dettagli di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata), vedere. [Visualizzazione dei dettagli di un gruppo di replica \(ElastiCache API\)](#)

## Visualizzazione di un gruppo di replica: Valkey o Redis OSS (modalità cluster abilitata)

### Visualizzazione di un cluster Valkey o Redis OSS (modalità cluster abilitata) (console)

Per visualizzare i dettagli di un cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando la ElastiCache console, vedere. [Visualizzazione dei dettagli per un cluster Valkey o Redis OSS \(Cluster Mode Enabled\) \(Console\)](#)

### Visualizzazione di un cluster Valkey o Redis OSS (modalità cluster abilitata) ()AWS CLI

Per un ElastiCache CLI esempio che mostra i dettagli di un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata), vedere. [Visualizzazione dei dettagli di un gruppo di replica \(AWS CLI\)](#)

### Visualizzazione di un cluster Valkey o Redis OSS (modalità cluster abilitata) () ElastiCache API

Per un ElastiCache API esempio che mostra i dettagli di un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata), vedere. [Visualizzazione dei dettagli di un gruppo di replica \(ElastiCache API\)](#)

### Visualizzazione dei dettagli di un gruppo di replica (AWS CLI)

È possibile visualizzare i dettagli di un gruppo di replica utilizzando il AWS CLI `describe-replication-groups` comando. 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 facoltativi

- `--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'output di questa operazione dovrebbe avere un aspetto simile JSON al seguente (formato).

```
{
 "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 AWS CLI l' ElastiCache argomento dedicato [describe-replication-groups](#).

## Visualizzazione dei dettagli di un gruppo di replica (ElastiCache API)

È possibile visualizzare i dettagli di una replica utilizzando l' AWS CLI

`DescribeReplicationGroups` operazione. 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 facoltativi

- `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, consultate l'argomento ElastiCache API di riferimento [DescribeReplicationGroups](#).

## Individuazione di endpoint del gruppo di replica

Un'applicazione può connettersi a qualsiasi nodo in un gruppo di replica, a condizione che disponga dell'endpoint e del numero di porta per quel nodo. A seconda che stiate utilizzando un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) o un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata), sarete interessati a diversi endpoint.

### Valkey o Redis (modalità cluster disabilitata) OSS

I cluster Valkey o Redis OSS (modalità cluster disabilitata) con repliche hanno tre tipi di endpoint: l'endpoint primario, l'endpoint reader e gli endpoint del nodo. L'endpoint primario è un DNS nome che si risolve sempre nel nodo primario del 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 suddividerà in modo uniforme le connessioni in entrata all'endpoint tra tutte le repliche di lettura in un cluster. ElastiCache 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. È possibile collocare le repliche di lettura multiple del cluster ElastiCache (RedisOSS) in diverse zone di disponibilità (AZ) per garantire l'elevata AWS disponibilità degli endpoint di lettura.

#### Note

Un endpoint di lettura non è un servizio di load balancer. È un DNS record che verrà risolto 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.

### Valkey o Redis OSS (modalità cluster abilitata)

I cluster Valkey o Redis OSS (modalità cluster abilitata) con repliche, poiché hanno più shard (API/CLI: gruppi di nodi), il che significa che hanno anche più nodi primari, hanno una struttura di endpoint diversa rispetto ai cluster Valkey o Redis (modalità cluster disabilitata). OSS Valkey o Redis OSS

(modalità cluster abilitata) dispone di un endpoint di configurazione che «conosce» tutti gli endpoint primari e i nodi del cluster. L'applicazione si connette all'endpoint di configurazione. Ogni volta che l'applicazione scrive o legge dall'endpoint di configurazione del cluster, Valkey e Redis, dietro le quinte OSS, determinano a quale shard appartiene la chiave e quale endpoint dello shard utilizzare. Tutto è trasparente all'applicazione.

È possibile trovare gli endpoint per un cluster utilizzando la console, il o il. ElastiCache AWS CLI  
ElastiCache API

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 Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)
- [Ricerca degli endpoint per un cluster \(console\) Valkey o Redis OSS \(modalità cluster abilitata\)](#)
- [Individuazione degli endpoint per i gruppi di OSS replica Valkey o Redis \(\)AWS CLI](#)
- [Ricerca degli endpoint per i gruppi di replica Valkey o Redis \(\) OSS ElastiCache API](#)

## Modifica di un gruppo di replica

### Vincoli importanti

- Attualmente, ElastiCache supporta modifiche limitate di un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata), ad esempio la modifica della versione del motore, utilizzando l'API operazione `ModifyReplicationGroup` (:). CLI `modify-replication-group` È possibile modificare il numero di shard (gruppi di nodi) in un cluster Valkey o Redis OSS (modalità cluster abilitata) con l'operazione (:). API [ModifyReplicationGroupShardConfigurationCLImodify-replication-group-shard-configuration](#) Per ulteriori informazioni, consulta [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#).

Altre modifiche a un cluster Valkey o Redis OSS (modalità cluster abilitata) richiedono la creazione di un cluster con il nuovo cluster che incorpori le modifiche.

- È possibile aggiornare i cluster e i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata) e Valkey o Redis OSS (modalità cluster abilitata) a versioni più recenti 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 [Gestione delle versioni per ElastiCache](#).
- È possibile aggiornare un cluster esistente ElastiCache con Valkey o Redis che utilizza la modalità OSS cluster disabilitata per utilizzare la modalità cluster abilitata, utilizzando la console [ModifyReplicationGroupAPI](#) o il comando, come illustrato nell'[modify-replication-group](#) CLI esempio seguente. Oppure, puoi seguire la procedura descritta in [Modifica della modalità cluster](#).

È possibile modificare le impostazioni di un cluster Valkey o Redis OSS (modalità cluster disabilitata) utilizzando la ElastiCache console, il AWS CLI o ElastiCache API. Attualmente, ElastiCache supporta un numero limitato di modifiche su un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata). Altre modifiche richiedono la creazione di un backup del gruppo di replica corrente e l'utilizzo di tale backup per avviare un nuovo gruppo di replica Valkey o Redis OSS (abilitata in modalità cluster).

### Argomenti

- [Usando il AWS Management Console](#)

- [Utilizzo del AWS CLI](#)
- [Usando il ElastiCache API](#)

## Usando il AWS Management Console

Per modificare un cluster Valkey o Redis OSS (modalità cluster disabilitata), vedere. [Modifica di un cluster ElastiCache](#)

## Utilizzo del AWS CLI

Di seguito sono AWS CLI riportati alcuni esempi del `modify-replication-group` comando. Puoi utilizzare lo stesso comando per apportare altre modifiche a un gruppo di replica.

Abilita Multi-AZ su un gruppo di replica Valkey o Redis OSS 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 OSS client Valkey o Redis di connettersi utilizzando sia la modalità cluster abilitata che la modalità cluster disabilitata. Dopo aver migrato tutti i OSS client Valkey o 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 \
 --multi-az-enabled
```

```
--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, vedere AWS CLI `modify-replication-group` [modify-replication-group](#) o [Modifica della modalità cluster](#) nella Guida per l'utente ElastiCache (RedisOSS).

Usando il ElastiCache API

La seguente ElastiCache API operazione abilita Multi-AZ su un gruppo di replica Valkey o Redis OSS 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, vedere ElastiCache API [ModifyReplicationGroup](#) [ModifyReplicationGroup](#).

## Eliminazione di un gruppo di replica

Se non hai più bisogno di uno dei tuoi cluster con repliche (chiamati gruppi di replica in API/CLI), puoi eliminarlo. Quando si elimina un gruppo di replica, ElastiCache vengono eliminati tutti i nodi di quel gruppo.

Dopo che è stata avviata, questa operazione non può essere interrotta o annullata.

### Warning

- Quando si elimina un cluster ElastiCache (RedisOSS), le istantanee manuali vengono conservate. Hai anche la possibilità di creare uno snapshot finale prima che il cluster venga eliminato. Gli snapshot di cache automatici non vengono conservati.
- `CreateSnapshot` è necessaria l'autorizzazione per creare un'istantanea finale. Senza questa autorizzazione, la API chiamata avrà esito negativo con un'Access Denied eccezione.

### Eliminazione di un gruppo di replica (console)

Per eliminare un cluster che dispone di repliche, consulta [Eliminazione di un cluster in ElastiCache](#).

### Eliminazione di un gruppo di replica (AWS CLI)

Usa 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 () ElastiCache API

Esegui una chiamata a [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

È possibile aumentare o diminuire dinamicamente il numero di repliche di lettura nel gruppo di OSS replica Valkey o Redis utilizzando il AWS Management Console, il o il. AWS CLI ElastiCache API. Se il gruppo di replica è un gruppo di replica Valkey o Redis OSS (abilitata in modalità cluster), è possibile scegliere quali shard (gruppi di nodi) aumentare o diminuire il numero di repliche.

Per modificare dinamicamente il numero di repliche nel gruppo di replica, scegliete l'operazione dalla tabella seguente che si adatta alla vostra situazione.

| Per                 | Per Valkey o Redis OSS (modalità cluster abilitata)            | Per Valkey o Redis OSS (modalità cluster disabilitata)                                                                                                                             |
|---------------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Aggiungere repliche | <a href="#">Incremento del numero di repliche in uno shard</a> | <a href="#">Incremento del numero di repliche in uno shard</a><br><br><a href="#">Aggiungere una replica di lettura per Valkey o Redis OSS (modalità cluster disabilitata)</a>     |
| Eliminare repliche  | <a href="#">Riduzione del numero di repliche in uno shard</a>  | <a href="#">Riduzione del numero di repliche in uno shard</a><br><br><a href="#">Eliminazione di una replica di lettura per Valkey o Redis OSS (modalità cluster disabilitata)</a> |

## Incremento del numero di repliche in uno shard

È possibile aumentare il numero di repliche in uno shard Valkey o Redis OSS (modalità cluster abilitata) o in un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) fino a un massimo di cinque. È possibile eseguire questa operazione utilizzando il, o il. AWS Management Console AWS CLI ElastiCache API

### Argomenti

- [Usando il AWS Management Console](#)
- [Usando il AWS CLI](#)
- [Usando il ElastiCache API](#)

### Usando il AWS Management Console

La procedura seguente utilizza la console per aumentare il numero di repliche in un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata).

Per aumentare il numero di repliche negli shard

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione, scegli Valkey o Redis OSS, quindi scegli il nome del gruppo di replica a cui desideri 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 Zone di disponibilità, scegli Nessuna preferenza per aver ElastiCache scelto una zona di disponibilità per ogni nuova replica oppure Specificare 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.

## Usando il AWS CLI

Per aumentare il numero di repliche in uno OSS shard Valkey o Redis, utilizzate il `increase-replica-count` comando 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` Facoltativo. Specifica il numero di nodi di replica desiderati, fino a un massimo di cinque. Utilizzate questo parametro per i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata) in cui è presente un solo gruppo di nodi o gruppo Valkey o Redis OSS (modalità cluster abilitata) o dove desiderate che tutti i gruppi di nodi abbiano 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` Facoltativo. Consente di impostare il numero di repliche e zone di disponibilità per ogni gruppo di nodi in maniera indipendente. Utilizzate questo parametro per i gruppi Valkey o Redis OSS (modalità cluster abilitata) in cui desiderate configurare ogni gruppo di nodi in modo 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 Valkey o Redis OSS (modalità cluster disabilitata), l'ID dello shard è sempre `0001`. Per trovare l'ID di un gruppo di nodi OSS (shard) di un gruppo di nodi Valkey o Redis (modalità cluster abilitata), vedi. [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 questo membro di `--replica-configuration` viene omesso, ElastiCache (RedisOSS) 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 si applica sia che si tratti di un gruppo Valkey o Redis OSS (modalità cluster disabilitata) con un gruppo di nodi singolo o di un gruppo Valkey o Redis OSS (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, si tratta di un gruppo di replica Valkey o Redis OSS (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 utilizzando il CLI, consulta [increase-replica-count](#) Amazon ElastiCache Command Line Reference.

Usando il ElastiCache API

Per aumentare il numero di repliche in uno OSS shard Valkey o Redis, utilizzate l'IncreaseReplicaCountazione 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** Facoltativo. Specifica il numero di nodi di replica desiderati, fino a un massimo di cinque. Utilizzate questo parametro per i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata) in cui è presente un solo gruppo di nodi o per i gruppi Valkey o Redis OSS (modalità cluster abilitata) in cui desiderate che tutti i gruppi di nodi abbiano 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` Facoltativo. Consente di impostare il numero di repliche e zone di disponibilità per ogni gruppo di nodi in maniera indipendente. Utilizzate questo parametro per i gruppi Valkey o Redis OSS (modalità cluster abilitata) in cui desiderate configurare ogni gruppo di nodi in modo indipendente.

`ReplicaConfiguraion` dispone di tre membri opzionali:

- `NodeId` - L'ID di quattro cifre del gruppo di nodi che si sta configurando. Per i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata), l'ID del gruppo di nodi (shard) è sempre `0001`. Per trovare l'ID di un gruppo di nodi OSS (shard) di un gruppo di nodi Valkey o Redis (modalità cluster abilitata), vedi [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 questo membro di `ReplicaConfiguration` viene omesso, ElastiCache (RedisOSS) 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 si applica sia che si tratti di un gruppo Valkey o Redis OSS (modalità cluster disabilitata) con un gruppo di nodi singolo o di un gruppo Valkey o Redis OSS (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, si tratta di un gruppo di replica Valkey o Redis OSS (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 utilizzando il API, consulta [IncreaseReplicaCount](#) Amazon ElastiCache API Reference.

## Riduzione del numero di repliche in uno shard

È possibile ridurre il numero di repliche in uno shard per Valkey o Redis OSS (modalità cluster abilitata) o in un gruppo di replica per Valkey o OSS Redis (modalità cluster disabilitata):

- Per Valkey o Redis OSS (modalità cluster disabilitata), è possibile ridurre il numero di repliche a una se Multi-AZ è abilitato e a zero se non è abilitato.
- Per Valkey o Redis OSS (modalità cluster abilitata), è possibile ridurre il numero di repliche a zero. Tuttavia, non puoi eseguire il failover in una replica se il nodo primario non riesce.

È possibile utilizzare il AWS Management Console, il AWS CLI o il ElastiCache API per ridurre il numero di repliche in un gruppo di nodi (shard) o in un gruppo di replica.

### Argomenti

- [Utilizzando il AWS Management Console](#)
- [Utilizzando il AWS CLI](#)
- [Usando il ElastiCache API](#)

### Utilizzando il AWS Management Console

La procedura seguente utilizza la console per ridurre il numero di repliche in un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata).

Per ridurre il numero di repliche in uno shard Valkey o Redis OSS

1. Accedi a AWS Management Console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Nel riquadro di navigazione, scegli Valkey o Redis OSS, quindi scegli il nome del gruppo di replica da cui desideri eliminare le 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 specificate i nodi di replica da eliminare, ElastiCache (RedisOSS) seleziona automaticamente i nodi di replica per l'eliminazione. Nel farlo, ElastiCache (RedisOSS) tenta di mantenere l'architettura Multi-AZ per il gruppo di replica, seguita da conservare le repliche con un ritardo di replica minimo con il primario.
- 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.


## Utilizzando il AWS CLI

Per ridurre il numero di repliche in uno OSS shard Valkey o Redis, utilizzate il `decrease-replica-count` comando 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` Facoltativo. 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` Facoltativo. Contiene un elenco di nodi che IDs specifica i nodi di replica da rimuovere.
- `--replica-configuration` Facoltativo. Consente di impostare il numero di repliche e zone di disponibilità per ogni gruppo di nodi in maniera indipendente. Utilizzate questo parametro per i gruppi Valkey o Redis OSS (modalità cluster abilitata) in cui desiderate configurare ogni gruppo di nodi in modo 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 Valkey o Redis OSS (modalità cluster disabilitata), l'ID dello shard è sempre `0001`. Per trovare l'ID di un gruppo di nodi OSS (shard) di un gruppo di nodi Valkey o Redis (modalità cluster abilitata), vedi [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 requisito, 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 questo membro di `--replica-configuration` viene omissso, ElastiCache (RedisOSS) 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 si applica sia che si tratti di un gruppo Valkey o Redis OSS (modalità cluster disabilitata) con un gruppo di nodi singolo o di un gruppo Valkey o Redis OSS (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, si tratta di un gruppo di replica Valkey o Redis OSS (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 utilizzando il CLI, consulta [decrease-replica-count](#) Amazon ElastiCache Command Line Reference.

Usando il ElastiCache API

Per ridurre il numero di repliche in uno OSS shard Valkey o Redis, utilizzate l'DecreaseReplicaCountazione 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** Facoltativo. 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** Facoltativo. Contiene un elenco di nodi che IDs specifica i nodi di replica da rimuovere.
- **ReplicaConfiguration** Facoltativo. 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. Utilizzate questo parametro per i gruppi Valkey o Redis OSS (modalità cluster abilitata) in cui desiderate configurare ogni gruppo di nodi in modo 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 Valkey o Redis OSS (modalità cluster disabilitata), l'ID del gruppo di nodi è sempre. 0001 Per trovare l'ID di un gruppo di nodi OSS (shard) di un gruppo di nodi Valkey o Redis (modalità cluster abilitata), vedi. [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 questo membro di ReplicaConfiguration viene omissso, ElastiCache (RedisOSS) 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 si applica sia che si tratti di un gruppo Valkey o Redis OSS (modalità cluster disabilitata) con un gruppo di nodi singolo o di un gruppo Valkey o Redis OSS (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, si tratta di un gruppo di replica Valkey o Redis OSS (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 utilizzando il API, consulta [Amazon DecreaseReplicaCount ElastiCache APIReference](#).

Aggiungere una replica di lettura per Valkey o Redis OSS (modalità cluster disabilitata)

Le informazioni contenute nel seguente argomento si applicano solo ai gruppi di replica Valkey o Redis OSS (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, puoi scoprire come aggiungere una replica di lettura a un cluster Valkey o Redis OSS (modalità cluster disabilitata).

Un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) può avere 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 Valkey o Redis OSS (modalità cluster abilitata), consulta quanto segue:

- [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)
- [Incremento del numero di repliche in uno shard](#)

È possibile aggiungere una replica di lettura a un cluster Valkey o Redis OSS (modalità cluster disabilitata) utilizzando Console, o. ElastiCache AWS CLI ElastiCache API

## Argomenti correlati

- [Aggiungere nodi a un ElastiCache cluster](#)
- [Aggiunta di una replica di lettura a un gruppo di replica \(AWS CLI\)](#)
- [Aggiungere una replica di lettura a un gruppo di replica utilizzando il 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 Valkey o Redis OSS (modalità cluster disabilitata), utilizzate il AWS CLI `create-cache-cluster` comando, con il parametro `--replication-group-id` per specificare a quale gruppo di replica 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 il, vedere CLI [create-cache-cluster](#) in Amazon ElastiCache Command Line Reference.

### Aggiungere una replica di lettura a un gruppo di replica utilizzando il API

Per aggiungere una replica di lettura a un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata), utilizzare l' `ElastiCache CreateCacheCluster` operazione, con il parametro `ReplicationGroupId` per specificare a quale gruppo di replica 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 il, vedere API [CreateCacheCluster](#) nell'Amazon ElastiCache API Reference.

Eliminazione di una replica di lettura per Valkey o Redis OSS (modalità cluster disabilitata)

Le informazioni contenute nel seguente argomento si applicano solo ai gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata).

Man mano che il traffico di lettura sul gruppo di OSS replica Valkey o Redis cambia, potresti voler aggiungere o rimuovere le repliche di lettura. La rimozione di un nodo da un gruppo di replica 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 a quella primaria, per i gruppi di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#).
  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.

È possibile rimuovere una replica di lettura da un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) utilizzando la console, il comando `for` o il ElastiCache . AWS CLI ElastiCache ElastiCache API

Per istruzioni sull'eliminazione di un cluster da un gruppo di replica Valkey o RedisOSS, consulta quanto segue:

- [Utilizzando il AWS Management Console](#)
- [Utilizzo di AWS CLI per eliminare un ElastiCache cluster](#)
- [Utilizzo del ElastiCache API](#)
- [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)
- [Riduzione del numero di repliche in uno shard](#)

## Promozione di una replica di lettura a quella primaria, per i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata)

Le informazioni riportate nel seguente argomento si applicano solo ai gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata).

È possibile promuovere una replica di lettura Valkey o Redis OSS (modalità cluster disabilitata) a replica di lettura principale utilizzando il, il AWS Management Console o. AWS CLI ElastiCache API Non puoi promuovere una replica di lettura al primario mentre la funzione Multi-AZ con failover automatico è abilitata sul gruppo di replica . Per promuovere una replica Valkey o Redis OSS (modalità cluster disabilitata) a principale su un gruppo di replica abilitato Multi-AZ, procedi come segue:

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.

Multi-AZ non è disponibile nei gruppi di replica che eseguono Redis 2.6.13 o versioni precedenti. OSS

Utilizzando il 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. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Se la replica che desideri promuovere è membro di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) in cui Multi-AZ è abilitato, modifica il gruppo di replica per disabilitare Multi-AZ prima di procedere. Per ulteriori informazioni, consulta [Modifica di un gruppo di replica](#).
3. Scegli Valkey o Redis OSS, quindi dall'elenco dei cluster, scegli il gruppo di replica che desideri 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](#).

## Utilizzando il 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 AWS CLI comando seguente modifica il gruppo di repliche `sample-repl-group`, rendendo la replica di lettura la principale `my-replica-1` del 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, vedere [modify-replication-group](#) in Amazon ElastiCache Command Line Reference.

### Utilizzando il ElastiCache API

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' ElastiCache APIazione seguente modifica il gruppo di replicamyRep1Group, rendendo la replica di lettura la principale myReplica-1 del gruppo di replica.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyReplicationGroup
&ReplicationGroupId=myReplGroup
&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 sulla modifica di un gruppo di replica, vedere [ModifyReplicationGroup](#) nell'Amazon ElastiCache API Reference.

## Gestione della manutenzione dei ElastiCache cluster

Ogni cluster ha una finestra di manutenzione settimanale durante la quale vengono applicate le modifiche al sistema. Con Valkey e RedisOSS, i gruppi di replica hanno la stessa finestra di manutenzione settimanale. Se non specifichi una finestra di manutenzione preferita quando crei o modifichi un cluster o un gruppo di replica, ElastiCache assegna una finestra di manutenzione di 60 minuti all'interno della finestra di manutenzione della regione in un giorno della settimana scelto a caso.

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-Local) | 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   | Medio Oriente ( ) Regione UAE                               | 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  | AWS GovCloud (US) regione                                   | 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 del cluster o del 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 desideri applicare immediatamente modifiche al tipo di nodo e/o aggiornamenti del motore, AWS Management Console seleziona la casella **Applica ora**. Altrimenti queste modifiche verranno applicate durante la successiva finestra di manutenzione programmata. Per utilizzare ilAPI, vedi [modify-replication-groupo](#). [modify-cache-cluster](#)

## Ulteriori informazioni

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

- [ElastiCache Manutenzione](#): FAQ sulla manutenzione e sulla sostituzione dei nodi
- [Sostituzione dei nodi \(Memcached\)](#)—Gestione della sostituzione dei nodi per Memcached
- [Modifica di un cluster ElastiCache](#) — Modifica della finestra di manutenzione di un cluster
- [Sostituzione dei nodi \(Valkey e RedisOSS\)](#):-Gestione della sostituzione dei nodi
- [Modifica di un gruppo di replica](#): modifica della finestra di manutenzione di un gruppo di replica

## Configurazione dei parametri del motore utilizzando gruppi di ElastiCache parametri

Amazon ElastiCache utilizza i parametri per controllare le proprietà di runtime dei nodi e dei cluster. Le versioni più recenti del prodotto includono in genere parametri aggiuntivi per il support delle nuove funzionalità. Per le tabelle dei parametri Memcached, vedi. [Parametri Memcached specifici](#) Per le tabelle dei parametri Valkey e RedisOSS, vedere. [Parametri Valkey e Redis OSS](#)

Come previsto, alcuni valori di parametro, ad esempio `maxmemory`, sono determinati da tipo di nodo e motore. Per una tabella di questi valori dei parametri Memcached per tipo di nodo, vedere. [Parametri specifici del tipo di nodo Memcached](#) Per una tabella di questi valori dei OSS parametri Valkey e Redis per tipo di nodo, vedere. [Parametri specifici del tipo di nodo Redis OSS](#)

### Note

Per un elenco dei parametri specifici Memcached, consultare [Parametri Memcached specifici](#).

### Argomenti

- [Gestione dei parametri in ElastiCache](#)
- [Memorizza i livelli del gruppo di parametri in ElastiCache](#)
- [Creazione di un gruppo di ElastiCache parametri](#)
- [Elenco ElastiCache dei gruppi di parametri per nome](#)
- [Elenco dei valori di un gruppo di ElastiCache parametri](#)
- [Modifica di un gruppo di ElastiCache parametri](#)
- [Eliminazione di un gruppo di ElastiCache parametri](#)
- [Parametri specifici del motore](#)



## Gestione dei parametri in ElastiCache

ElastiCache i parametri sono raggruppati in gruppi di parametri denominati per una più semplice gestione dei parametri. 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.
- Per Memcached, la famiglia di gruppi di parametri e il cluster a cui lo stai assegnando devono essere compatibili. Se il cluster esegue ad esempio Memcached versione 1.4.8, puoi usare solo gruppi di parametri della famiglia Memcached 1.4, predefiniti o personalizzati.

Per Redis, la famiglia di gruppi di parametri e il cluster a cui la stai assegnando devono essere compatibili. Ad esempio, se sul cluster è in esecuzione la OSS versione 3.2.10 di Redis, è possibile utilizzare solo gruppi di parametri, predefiniti o personalizzati, della famiglia Redis 3.2. OSS

- 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.
- Per Memcached, quando si modificano i parametri di un cluster, la modifica viene applicata immediatamente al 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 un determinato parametro, consulta la colonna relativa all'applicazione delle modifiche nelle tabelle in [Parametri Memcached specifici](#). Per ulteriori informazioni sul riavvio dei nodi del cluster, consulta [Rebooting clusters](#).
- Per Redis, quando si modificano i parametri di un cluster, la modifica viene applicata al cluster immediatamente o, con le eccezioni riportate di seguito, dopo il riavvio dei nodi 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 un determinato parametro, consulta la colonna relativa all'applicazione delle modifiche nelle tabelle in [Parametri Valkey e Redis OSS](#).

Per ulteriori informazioni sul riavvio dei nodi Valkey o Redis, consulta. OSS [Riavvio dei nodi](#)

### Modifiche ai parametri Valkey o Redis OSS (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri su un cluster Valkey o Redis OSS (modalità cluster abilitata), segui i passaggi seguenti.

- activerehashing
  - database
1. Creare un backup manuale del cluster. Per informazioni, consulta [Esecuzione di backup manuali](#).
  2. Elimina il cluster. Consulta [Deleting clusters](#).
  3. Ripristina il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consulta [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

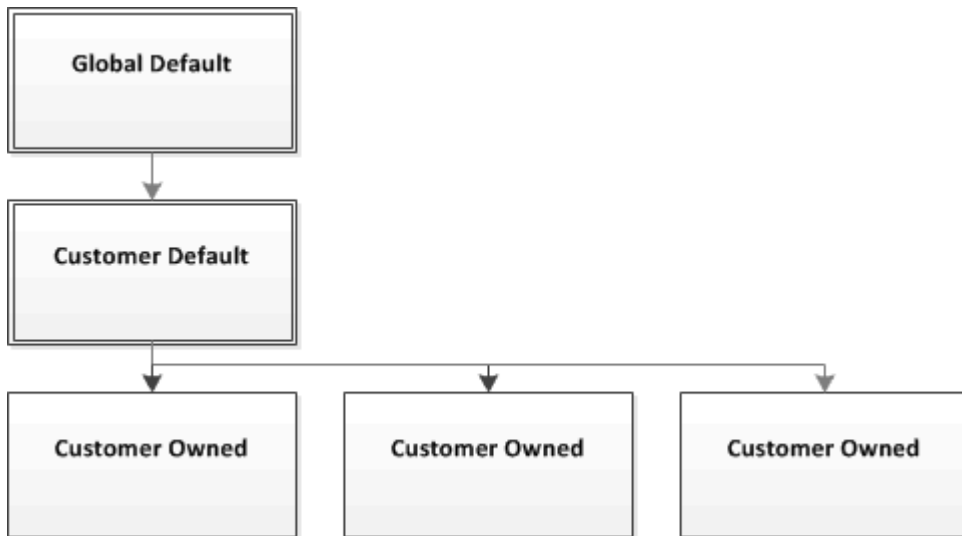
- È possibile associare gruppi di parametri ai datastore globali Valkey e Redis. OSS I datastore globali sono una raccolta di uno o più cluster che si estendono su più regioni. AWS In questo caso, il gruppo di parametri viene condiviso da tutti i cluster che costituiscono il Global Datastore. Eventuali modifiche al gruppo di parametri del cluster primario vengono replicate in tutti i cluster rimanenti nel Global Datastore. Per ulteriori informazioni, consulta [Replica tra AWS regioni utilizzando datastore globali](#).

È possibile verificare se un gruppo di parametri fa parte di un Global Datastore esaminando le seguenti posizioni:

- Nella ElastiCache console, nella pagina Parameter Groups, l'attributo Global yes/no
- La proprietà yes/no IsGlobal dell'operazione [CacheParameterGroupAPI](#)

## Memorizza i livelli del gruppo di parametri in ElastiCache

Amazon ElastiCache dispone di tre livelli di gruppi di parametri di cache, come illustrato di seguito.



### Livelli ElastiCache dei gruppi di parametri Amazon

#### Di default globale

Il gruppo di parametri principali di primo livello per tutti ElastiCache i clienti Amazon della regione.

Gruppo di parametri cache di default globale:

- È riservato 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:

- È stato creato e di proprietà 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 sui valori Memcached, vedere. [Parametri Memcached specifici](#) Per ulteriori informazioni sui valori Valkey e RedisOSS, vedere. [Parametri Valkey e Redis OSS](#)

## Creazione di un gruppo di ElastiCache parametri

Devi creare un nuovo gruppo di parametri se per uno o più valori di parametri desideri configurare un'impostazione diversa da quella predefinita. È possibile creare un gruppo di parametri utilizzando la ElastiCache console, il AWS CLI, o il ElastiCache API.

### Creazione di un gruppo di ElastiCache parametri (Console)

La procedura seguente mostra come creare un gruppo di parametri utilizzando la ElastiCache console.

Per creare un gruppo di parametri utilizzando la ElastiCache console

1. Accedi a AWS Management Console e apri la ElastiCache console 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 di gruppi di parametri, come `memcached1.4` o `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:

- Deve iniziare con una lettera. ASCII
  - Può contenere solo ASCII lettere, 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 ElastiCache parametri](#).

## Creazione di un gruppo di ElastiCache parametri (AWS CLI)

Per creare un gruppo di parametri utilizzando il AWS CLI, utilizzate 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:

- Deve iniziare con una ASCII lettera.
- Può contenere solo ASCII lettere, 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 gruppo di parametri denominato `myMem14` utilizzando la famiglia `memcached1.4` come modello.

Per Linux, macOS o Unix:

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name myMem14 \
 --cache-parameter-group-family memcached1.4 \
 --description "My first parameter group"
```

Per Windows:

```
aws elasticache create-cache-parameter-group ^
 --cache-parameter-group-name myMem14 ^
 --cache-parameter-group-family memcached1.4 ^
 --description "My first parameter group"
```

L'output di questo comando dovrebbe essere simile a quanto segue:

```
{
 "CacheParameterGroup": {
 "CacheParameterGroupName": "myMem14",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
}
```

## Example

L'esempio seguente crea un gruppo di parametri denominato `myRed28` utilizzando 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 ElastiCache parametri](#).

Per ulteriori informazioni, consulta [create-cache-parameter-group](#).

Creazione di un gruppo di ElastiCache parametri () ElastiCache API

Per creare un gruppo di parametri utilizzando ElastiCache API, utilizzate l'CreateCacheParameterGroupazione con questi parametri.

- ParameterGroupName - Nome del gruppo di parametri.

Vincoli per la denominazione dei gruppi di parametri:

- Deve iniziare con una ASCII lettera.
- Può contenere solo ASCII lettere, 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 `memcached1.4`.
- `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 gruppo di parametri denominato `myMem14` utilizzando la famiglia `memcached1.4` come modello.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheParameterGroup
&CacheParameterGroupFamily=memcached1.4
&CacheParameterGroupName=myMem14
&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>myMem14</CacheParameterGroupName>
 <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
 <Description>My first parameter group</Description>
 </CacheParameterGroup>
 </CreateCacheParameterGroupResult>
 <ResponseMetadata>
 <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
 </ResponseMetadata>
```

```
</CreateCacheParameterGroupResponse>
```

## Example

L'esempio seguente crea un gruppo di parametri denominato myRed28 utilizzando 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 ElastiCache parametri](#).

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

## Elenco ElastiCache dei gruppi di parametri per nome

È possibile elencare i gruppi di parametri utilizzando la ElastiCache console, il AWS CLI, o il ElastiCacheAPI.

### Elenco di gruppi di parametri per nome (console)

La procedura seguente mostra come visualizzare un elenco dei gruppi di parametri utilizzando la ElastiCache console.

Per elencare i gruppi di parametri utilizzando la ElastiCache console

1. Accedi a AWS Management Console e apri la ElastiCache console 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 ElastiCache dei gruppi di parametri per nome (AWS CLI)

Per generare un elenco di gruppi di parametri utilizzando il AWS CLI, utilizzate il comandodescribe-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 myMem14.

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myMem14
```

Per Windows:

```
aws elasticache describe-cache-parameter-groups ^
 --cache-parameter-group-name myMem14
```

L'output di questo comando sarà simile al seguente e conterrà il nome, la famiglia e la descrizione del gruppo di parametri.

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myMem14",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
]
}
```

## 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 codice di esempio seguente elenca il gruppo di parametri myRed56 per i gruppi di parametri in esecuzione sul OSS motore Redis dalla versione 5.0.6 in poi. Se il gruppo di parametri fa parte di

un [Replica tra AWS regioni utilizzando datastore globali](#), 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, con l'elenco del nome, della famiglia `isGlobal` e della 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'JSONOutput di questo comando sarà simile al seguente, elencando il nome, la famiglia, la descrizione e, nel caso di `redis5.6`, se il gruppo di parametri fa parte di un datastore globale (`isGlobal`), per ogni gruppo di parametri.

```
{
 "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 ElastiCache parametri per nome () ElastiCache API

Per generare un elenco di gruppi di parametri utilizzando il ElastiCache API, utilizzate l'DescribeCacheParameterGroupsazione. 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 il gruppo di parametri myMem14.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myMem14
&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 di ciascun gruppo di parametri.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/
doc/2013-06-15/">
 <DescribeCacheParameterGroupsResult>
 <CacheParameterGroups>
 <CacheParameterGroup>
 <CacheParameterGroupName>myMem14</CacheParameterGroupName>
 <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
 <Description>My custom Memcached 1.4 parameter group</Description>
 </CacheParameterGroup>
 </CacheParameterGroups>
 </DescribeCacheParameterGroupsResult>
 <ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

## 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 azione sarà simile a questa, elencando il nome, la famiglia, la descrizione e, nel caso di redis5.6 se il gruppo di parametri appartiene a un datastore globale (isGlobal), per ogni gruppo di parametri.

```
<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 28. *myRed*

```
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 azione sarà simile alla seguente, con l'elenco del nome, della famiglia, della 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>
 <ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

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

## Elenco dei valori di un gruppo di ElastiCache parametri

È possibile elencare i parametri e i relativi valori per un gruppo di parametri utilizzando la ElastiCache console, il AWS CLI, o il ElastiCache API.

### Elenco dei valori di un gruppo di ElastiCache parametri (Console)

La procedura seguente mostra come elencare i parametri e i relativi valori per un gruppo di parametri utilizzando la ElastiCache console.

Per elencare i parametri di un gruppo di parametri e i relativi valori utilizzando la ElastiCache console

1. Accedi AWS Management Console e apri la ElastiCache console 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 elencare i parametri di un gruppo di parametri e i relativi valori utilizzando il AWS CLI, usa il comando `describe-cache-parameters`.

### Example

Il codice di esempio seguente elenca tutti i parametri Memcached e i relativi valori per il gruppo myMem di parametri 14.

Per Linux, macOS o Unix:

```
aws elasticache describe-cache-parameters \
 --cache-parameter-group-name myMem14
```

Per Windows:

```
aws elasticache describe-cache-parameters ^
 --cache-parameter-group-name myMem14
```

## Example

Il codice di esempio seguente elenca tutti i parametri e i relativi valori per il gruppo myRedis di parametri 28.

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 (ElastiCache API)

Per elencare i parametri di un gruppo di parametri e i relativi valori utilizzando il ElastiCache API, utilizzate l'DescribeCacheParametersazione.

## Example

Il codice di esempio seguente elenca tutti i parametri Memcached per il gruppo myMem di parametri 14.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameters
&CacheParameterGroupName=myMem14
&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>
 </CacheClusterClassSpecificParameters>
 </DescribeCacheParametersResult>
 <ResponseMetadata>
 <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
 </DescribeCacheParametersResponse>

```

...output omitted...

## Example

Il codice di esempio seguente elenca tutti i parametri per il gruppo myRed di parametri 28.

```

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>
 </CacheNodeTypeSpecificValues>
 </CacheClusterClassSpecificParameters>
 </DescribeCacheParametersResult>
 <ResponseMetadata>

```

...output omitted...



```
<RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParametersResponse>
```

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

## Modifica di un gruppo di ElastiCache 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 del valore di un parametro viene applicata a un gruppo di parametri, vedere [Parametri Valkey e Redis OSS e Parametri Memcached specifici](#).

### Modifica di un gruppo di parametri (console)

La procedura seguente mostra come modificare il valore del `cluster-enabled` parametro utilizzando la ElastiCache console. Puoi usare la stessa procedura per modificare il valore di qualsiasi parametro.

Per modificare il valore di un parametro utilizzando la ElastiCache console


1. Accedi a AWS Management Console e apri la ElastiCache console 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. Nella schermata Edit Parameter Group: (Modifica il gruppo di parametri:) scorrere utilizzando le frecce sinistra e destra fino al parametro `binding_protocol`, quindi digitare `ascii` nella colonna Value (Valore).

6. Seleziona Salva modifiche.
7. Per Memcached, per trovare il nome del parametro modificato, vedi. [Parametri Memcached specifici](#) Se le modifiche apportate al parametro vengono applicate dopo il riavvio, riavvia ogni cluster che utilizza il gruppo di parametri. Per ulteriori informazioni, consulta [Rebooting clusters](#).
8. Con Valkey e RedisOSS, per trovare il nome del parametro modificato, vedi. [Parametri Valkey e Redis OSS](#) Se disponi di un cluster Valkey o Redis OSS (modalità cluster disabilitata) e apporti modifiche ai seguenti parametri, devi riavviare i nodi del cluster:
  - activerehashing
  - database

Per ulteriori informazioni sul riavvio, consultare [Riavvio dei nodi](#).

 Modifiche ai parametri Valkey o Redis OSS (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri su un cluster Valkey o Redis OSS (modalità cluster abilitata), segui i passaggi seguenti.

- activerehashing
- database

1. Con Redis, puoi creare un backup manuale del tuo cluster. Per informazioni, consulta [Esecuzione di backup manuali](#).
2. Elimina il cluster. Consulta [Deleting clusters](#).
3. Ripristinare il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consulta [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 utilizzando il AWS CLI, usa il comando. `modify-cache-parameter-group`

## Example

Con Memcached, per trovare il nome e i valori consentiti del parametro che desideri modificare, vedi [Parametri Memcached specifici](#)

Il seguente codice di esempio imposta il valore di due parametri, `chunk_size` e `chunk_size_growth_fact`, nel gruppo di parametri `myMem14`.

Per Linux, macOS o Unix:

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name myMem14 \
 --parameter-name-values \
 ParameterName=chunk_size,ParameterValue=96 \
 ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Per Windows:

```
aws elasticache modify-cache-parameter-group ^
 --cache-parameter-group-name myMem14 ^
 --parameter-name-values ^
 ParameterName=chunk_size,ParameterValue=96 ^
 ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

L'aspetto dell'output di questo comando sarà simile al seguente.

```
{
 "CacheParameterGroupName": "myMem14"
}
```

## Example

Con Valkey e RedisOSS, per trovare il nome e i valori consentiti del parametro che desideri modificare, vedi [Parametri Valkey e Redis OSS](#)

Il codice di esempio seguente imposta il valore di due parametri `reserved-memory-percent` e `reserved-memory-percent-per-cluster` abilitato in cluster sul gruppo di parametri. `myredis32-on-30` Abbiamo impostato su `30` (30 per cento) e abilitato `reserved-memory-percent-per-cluster` per il cluster in `yes` modo che il gruppo di parametri possa essere utilizzato con i cluster (gruppi di replica) Valkey o Redis OSS (abilitati per la modalità cluster).

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 Valkey e Redis OSS](#).

Se disponi di un cluster Valkey o Redis OSS (modalità cluster disabilitata) e apporti modifiche ai seguenti parametri, devi riavviare i nodi del cluster:

- activerehashing
- database

Per ulteriori informazioni sul riavvio, consultare [Riavvio dei nodi](#).

#### Modifiche ai parametri Valkey o Redis OSS (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri su un cluster Valkey o Redis OSS (modalità cluster abilitata), segui i passaggi seguenti.

- activerehashing
- database

1. Creare un backup manuale del cluster. Per informazioni, consulta [Esecuzione di backup manuali](#).
2. Elimina il cluster. Consulta [Deleting clusters](#).
3. Ripristinare il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consulta [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

## Modifica di un gruppo di parametri (ElastiCache API)

Per modificare i valori dei parametri di un gruppo di parametri utilizzando il ElastiCache API, utilizzate l'azione. `ModifyCacheParameterGroup`

### Example

Con Memcached, per trovare il nome e i valori consentiti del parametro che desideri modificare, vedi [Parametri Memcached specifici](#)

Il seguente codice di esempio imposta il valore di due parametri, `chunk_size` e `chunk_size_growth_fact`, nel gruppo di parametri `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheParameterGroup
&CacheParameterGroupName=myMem14
&ParameterNameValues.member.1.ParameterName=chunk_size
&ParameterNameValues.member.1.ParameterValue=96
&ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact
&ParameterNameValues.member.2.ParameterValue=1.5
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

## Example

Con Valkey e RedisOSS, per trovare il nome e i valori consentiti del parametro che desideri modificare, vedi [Parametri Valkey e Redis OSS](#)

Il codice di esempio seguente imposta il valore di due parametri `reserved-memory-percent` e `cluster-enabled` in cluster sul gruppo di parametri `myredis32-on-30`. Abbiamo impostato su `30` (30 per cento) e abilitato `reserved-memory-percent` per il cluster in `yes` modo che il gruppo di parametri possa essere utilizzato con i cluster (gruppi di replica) Valkey o Redis OSS (abilitati per la modalità cluster).

```
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 disponi di un cluster Valkey o Redis OSS (modalità cluster disabilitata) e apporti modifiche ai seguenti parametri, devi riavviare i nodi del cluster:

- `activeresharding`
- `database`

Per ulteriori informazioni, consulta [Riavvio dei nodi](#).

### Modifiche ai parametri Valkey o Redis OSS (modalità cluster abilitata)

Se apporti modifiche ai seguenti parametri su un cluster Valkey o Redis OSS (modalità cluster abilitata), segui i passaggi seguenti.

- `activeresharding`

- database

1. Creare un backup manuale del cluster. Per informazioni, consulta [Esecuzione di backup manuali](#).
2. Elimina il cluster. Per informazioni, consulta [Eliminazione di un cluster in ElastiCache](#).
3. Ripristinare il cluster utilizzando il gruppo di parametri modificato e il backup per inizializzare il nuovo cluster. Per informazioni, consulta [Ripristino da un backup in una nuova cache](#).

Le modifiche ad altri parametri non richiedono questo.

## Eliminazione di un gruppo di ElastiCache parametri

È possibile eliminare un gruppo di parametri personalizzato utilizzando la ElastiCache console, il AWS CLI, o il ElastiCache API.

Un gruppo di parametri non può essere eliminato se è associato a cluster. Non è inoltre possibile eliminare i gruppi di parametri predefiniti.

### Eliminazione di un gruppo di parametri (console)

La procedura seguente mostra come eliminare un gruppo di parametri utilizzando la ElastiCache console.

Per eliminare un gruppo di parametri utilizzando la ElastiCache console

1. Accedi a AWS Management Console e apri la ElastiCache console 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. Scegli 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 utilizzando il AWS CLI, utilizzare 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 myMem14 parametri.

### 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 (ElastiCache API)

Per eliminare un gruppo di parametri utilizzando il ElastiCache API, utilizzate l>DeleteCacheParameterGroupazione. 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

Con Memcached, il seguente codice di esempio elimina il gruppo di myMem14 parametri.

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=DeleteCacheParameterGroup
 &CacheParameterGroupName=myMem14
 &SignatureVersion=4
 &SignatureMethod=HmacSHA256
 &Timestamp=20150202T192317Z
 &Version=2015-02-02
 &X-Amz-Credential=<credential>
```

### Example

Il codice di esempio seguente elimina il gruppo di 28 parametri. myRed

```
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 specifici del motore

### Valkey e Redis OSS

I parametri di Valkey 7.2 sono gli stessi dei parametri di Redis 7. OSS

Se non si specifica un gruppo di parametri per il OSS cluster Valkey o Redis, verrà utilizzato un gruppo di parametri predefinito appropriato alla 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 ElastiCache parametri](#).

### Argomenti

- [Parametri Valkey e Redis OSS](#)
- [Parametri Memcached specifici](#)

## Parametri Valkey e Redis OSS

### Argomenti

- [Modifiche ai parametri di Valkey 7.2 e Redis 7 OSS](#)
- [Modifiche ai parametri di Redis OSS 6.x](#)
- [Modifiche ai parametri di Redis 5.0.3 OSS](#)
- [Modifiche ai parametri di Redis 5.0.0 OSS](#)
- [Modifiche ai parametri Redis OSS 4.0.10](#)
- [Modifiche ai parametri Redis 3.2.10 OSS](#)
- [Modifiche ai parametri Redis 3.2.6 OSS](#)
- [Modifiche ai parametri di Redis 3.2.4 OSS](#)
- [Parametri aggiunti per Redis OSS 2.8.24 \(migliorato\)](#)
- [Parametri aggiunti per Redis OSS 2.8.23 \(migliorato\)](#)
- [Parametri aggiunti Redis OSS 2.8.22 \(migliorati\)](#)
- [Parametri aggiunti da Redis OSS 2.8.21](#)
- [Parametri aggiunti Redis 2.8.19 OSS](#)
- [Parametri aggiunti da Redis 2.8.6 OSS](#)
- [Parametri Redis 2.6.13 OSS](#)
- [Parametri specifici del tipo di nodo Redis OSS](#)

### Modifiche ai parametri di Valkey 7.2 e Redis 7 OSS

Famiglia del gruppo di parametri: redis7

I gruppi di parametri predefiniti di Redis OSS 7 sono i seguenti:

- `default.redis7`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per i cluster e i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata).
- `default.redis7.cluster.on`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per cluster e gruppi di replica Valkey o Redis OSS (modalità cluster abilitata).

I parametri aggiunti in Redis OSS 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 <code>MOVED/ASKING</code> requests e il campo <code>endpoint</code> per <code>and</code>. <code>CLUSTER SLOTS</code> <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 pubblicizzerà un nome host quando <code>encryption-in-transit</code> è abilitato 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>Modificabile: sì</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 di latenza cumulative (istogrammi) tramite il comando <code>LATENCY</code>.</p>

Nome	Informazioni	Descrizione
	<p>Le modifiche diventano effettive: immediatamente in tutti i nodi del cluster.</p>	
<p>hash-max-listpack-entries</p>	<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>
<p>hash-max-listpack-value</p>	<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 OSS 7 sono i seguenti.

Nome	Informazioni	Descrizione
<code>activereshashing</code>	<p>Modificabile: no. In Redis OSS 7, questo parametro è nascosto e abilitato per impostazione predefinita. Per disattivarlo, è necessario creare un <a href="#">caso di supporto</a>.</p>	Era modificabile.

I parametri rimossi in Redis OSS 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 listpack anziché ziplist 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 listpack anziché ziplist 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 listpack anziché ziplist 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 ai parametri di Redis OSS 6.x

Famiglia del gruppo di parametri: `redis6.x`

I gruppi di parametri predefiniti di Redis OSS 6.x sono i seguenti:

- `default.redis6.x`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per i cluster e i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata).
- `default.redis6.x.cluster.on`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per cluster e gruppi di replica Valkey o Redis OSS (modalità cluster abilitata).

### Note

Nella versione 6.2 OSS del motore Redis, quando è stata introdotta la famiglia di nodi r6gd per l'uso con, solo le politiche di memoria massima noeviction [Suddivisione dei dati su più livelli in ElastiCache](#), volatile-lru e allkeys-lru sono supportate con i tipi di nodi r6gd.

Per ulteriori informazioni, consulta [ElastiCache \(Redis\) versione 6.2 \(migliorataOSS\)](#) e [ElastiCache \(Redis\) versione 6.0 \(migliorataOSS\)](#).

I parametri aggiunti in Redis OSS 6.x sono i seguenti.

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 hanno effetto: gli OSS utenti Redis esistenti associati al cluster continueranno ad avere le autorizzazioni esistenti. Aggiorna gli utenti o riavvia il cluster per aggiornare gli utenti Redis esistenti. OSS</p>	Autorizzazioni predefinite del canale pubsub per ACL gli utenti distribuiti 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 <code>yes</code> , un gruppo di replica Redis OSS (modalità cluster abilitata) continua a elaborare i comandi di lettura anche quando un nodo non è in grado di raggiungere un quorum di primari.

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 facilitare la memorizzazione nella cache lato client, Redis supporta il monitoraggio dei client che hanno effettuato l'accesso a quali chiavi OSS.</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

Informazioni	Descrizione	
<p><code>active-expire-effort</code> (added in 6.0)</p>	<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 OSS elimina le chiavi che hanno superato il loro periodo di validità mediante 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 lo sforzo impiegato da Redis OSS per far scadere gli elementi del job 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 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 latenza più alta CPU e potenzialmente una maggiore. Consigliamo un valore pari a 1 a meno che non si riscontri un utilizzo elevato della memoria e si possa tollerare un aumento dell'utilizzo. CPU</p>
<p><code>lazyfree-lazy-user-del</code> (added in 6.0)</p>	<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 OSS 6.x sono i seguenti.

Nome	Informazioni	Descrizione
lua-replicate-commands	<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>

### Modifiche ai parametri di Redis 5.0.3 OSS

Famiglia del gruppo di parametri: redis5.0

Gruppi di parametri predefiniti di Redis OSS 5.0

- `default.redis5.0`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per i cluster e i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata).
- `default.redis5.0.cluster.on`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per cluster e gruppi di replica Valkey o Redis OSS (modalità cluster abilitata).

### Parametri aggiunti in Redis 5.0.3 OSS

Nome	Informazioni	Descrizione
rename-commands	<p>Impostazione predefinita: none</p> <p>Tipo: stringa</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive immediatamente in tutti i nodi del cluster</p>	<p>Un elenco separato da spazi di comandi Redis rinominati. OSS Di seguito è riportato un elenco limitato di comandi disponibili per la ridenominazione:</p> <p>APPEND AUTH BITCOUNT BITFIELD            BITOP BITPOS BLPOP BRPOP BR            POPLUSH BZPOPMIN BZPOPMAX CLIENT            CLUSTER COMMAND DBSIZE DECR</p>

Nome	Informazioni	Descrizione
		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

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 \(Redis\) versione 5.0.6 \(migliorataOSS\)](#).

Modifiche ai parametri di Redis 5.0.0 OSS

Famiglia del gruppo di parametri: redis5.0

Gruppi di parametri predefiniti di Redis 5.0 OSS

- `default.redis5.0`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per i cluster e i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata).
- `default.redis5.0.cluster.on`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per cluster e gruppi di replica Valkey o Redis OSS (modalità cluster abilitata).

Parametri aggiunti in Redis 5.0 OSS

Nome	Informazioni	Descrizione
<code>stream-node-max-bytes</code>	<p>Valori consentiti: 0+</p> <p>Impostazione predefinita: 4096</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 le dimensioni massime in byte di un singolo nodo in una struttura radice. Se impostata su 0, il nodo della struttura è illimitato.</p>

Nome	Informazioni	Descrizione
stream-node-max-entries	<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>
active-defrag-max-scan-fields	<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>
lua-replicate-commands	<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 OSS ha rinominato diversi parametri nella versione 5.0 del motore in risposta al feedback della community. Per ulteriori informazioni, consulta [Cosa c'è di nuovo in Redis OSS 5?](#) . La tabella seguente elenca i nuovi nomi e il modo in cui vengono mappati alle versioni precedenti.

#### Parametri rinominati in Redis 5.0 OSS

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 <a href="#">Parametri specifici del tipo di nodo Redis OSS</a>  Tipo: Integer  Modificabile: no  Nome precedente: <code>client-output-buffer-limit - slave-hard-limit</code>	Per le repliche di OSS 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 <a href="#">Parametri specifici del tipo di nodo Redis OSS</a></p> <p>Tipo: Integer</p> <p>Modificabile: no</p> <p>Nome precedente: - client-output-buffer-limit slave-soft-limit</p>	Per le repliche di OSS 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. <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: - client-output-buffer-limit slave-soft-seconds</p>	Per le repliche di OSS lettura Redis: se il buffer di output di un client rimane in <code>client-output-buffer-limit-replica-soft-limit</code> byte per un periodo superiore a questo numero di secondi, 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: slave-allow-chaining</p>	Determina se una replica di lettura in Redis OSS può avere repliche di lettura proprie.

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: min-slaves-to-write</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 primario 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: min-slaves-max-lag</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 <code>min-replicas-to-write</code>, a quel punto il primario smetterà di accettare scritture.</p> <p>Se questo parametro o <code>min-replicas-to-write</code> è 0, il nodo primario accetterà sempre le richieste di scrittura, anche se non sono disponibili repliche.</p>

Nome	Informazioni	Descrizione
<code>close-on-replica-write</code>	Impostazione predefinita: yes (sì) Tipo: booleano Modificabile: sì Nome precedente: close-on-slave-write 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 OSS

Nome	Informazioni	Descrizione
<code>repl-timeout</code>	Impostazione predefinita: 60 Modificabile: no	Il parametro non è disponibile in questa versione.

### Modifiche ai parametri Redis OSS 4.0.10

Famiglia del gruppo di parametri: `redis4.0`

### Gruppi di parametri predefiniti di Redis 4.0.x OSS

- `default.redis4.0`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per i cluster e i gruppi di replica Valkey o Redis OSS (modalità cluster disabilitata).
- `default.redis4.0.cluster.on`— Utilizzate questo gruppo di parametri, o uno da esso derivato, per cluster e gruppi di replica Valkey o Redis OSS (modalità cluster abilitata).

## Parametri modificati in Redis 4.0.10 OSS

Nome	Informazioni	Descrizione
maxmemory-policy	<p>Valori consentiti: allkeys-lru , volatile-lru , <b>allkeys-lfu</b> , <b>volatile-lfu</b> , 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 vengono aggiunti due nuovi valori consentiti: allkeys-lfu , che eliminerà qualsiasi chiave utilizzando approssimated e LFUvolatile-lfu , che eliminerà utilizzando approssimated tra le chiavi con una scadenza impostata. LFU 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 e con i tipi di nodi r6gd.</p>

## Parametri aggiunti in Redis 4.0.10 OSS

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>Esegue un'eliminazione asincrona sulle espulsioni.</p>

Nome	Informazioni	Descrizione
lazyfree-lazy-expire	<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>	Esegue un'eliminazione asincrona sulle chiavi scadute.
lazyfree-lazy-server-del	<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>	Esegue un'eliminazione asincrona per i comandi che aggiornano i valori.
slave-lazy-flush	<p>Valori consentiti: N/D</p> <p>Impostazione predefinita: no</p> <p>Tipo: booleano</p> <p>Modificabile: no</p> <p>Le modifiche diventano effettive: N/D</p>	Esegue un'operazione flushDB asincrona durante la sincronizzazione slave.
LFUparametri		

Nome	Informazioni	Descrizione
<code>lfu-log-factor</code>	<p>Valori consentiti: qualsiasi numero intero &gt; 0</p> <p>Impostazione predefinita: 10</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente.</p>	<p>Imposta il fattore di log, che determina il numero di occorrenze delle chiavi per saturare il contatore delle chiavi.</p>
<code>lfu-decay-time</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>La quantità di tempo in minuti per diminuire il contatore delle chiavi.</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>Deframmentazione attiva abilitata.</p>

Nome	Informazioni	Descrizione
<code>active-defrag-ignore-bytes</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>	Quantità minima di scarto della frammentazione necessari a per avviare la deframmentazione attiva.
<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>	Percentuale minima di frammentazione necessari a per avviare la deframmentazione attiva.
<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>	Percentuale massima di frammentazione che richiede lo sforzo massimo.



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.	Minimo sforzo di deframmentazione in CPU percentuale.
<code>active-defrag-cycle-max</code>	Valori consentiti: 1-75  Impostazione predefinita: 75  Tipo: Integer  Modificabile: sì  Le modifiche diventano effettive: immediatamente.	Sforzo massimo per la deframmentazione in percentuale. CPU
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.	Dimensione massima di un singolo buffer di query client.

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>	Dimensione massima di una singola richiesta di elementi.

### Modifiche ai parametri Redis 3.2.10 OSS

Famiglia del gruppo di parametri: `redis3.2`

ElastiCache (RedisOSS) 3.2.10 non sono supportati parametri aggiuntivi.

### Modifiche ai parametri Redis 3.2.6 OSS

Famiglia del gruppo di parametri: `redis3.2`

Per Redis OSS 3.2.6 non sono supportati parametri aggiuntivi.

### Modifiche ai parametri di Redis 3.2.4 OSS

Famiglia del gruppo di parametri: `redis3.2`

A partire da Redis OSS 3.2.4 ci sono due gruppi di parametri predefiniti.

- `default.redis3.2`— Quando esegui Redis OSS 3.2.4, specifica questo gruppo di parametri o uno da esso derivato, se desideri creare un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) e continuare a utilizzare le funzionalità aggiuntive di Redis 3.2.4. OSS
- `default.redis3.2.cluster.on`— Specificate questo gruppo di parametri o uno da esso derivato, quando desiderate creare un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata).

## Argomenti

- [Nuovi parametri per Redis 3.2.4 OSS](#)
- [Parametri modificati in Redis OSS 3.2.4 \(migliorato\)](#)

### Nuovi parametri per Redis 3.2.4 OSS

Famiglia del gruppo di parametri: redis3.2

Per Redis OSS 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"><li>• -5: dimensione massima: 64 KB (non consigliato per carichi di lavoro normali)</li><li>• -4: dimensione massima: 32 KB (non consigliato)</li><li>• -3: dimensione massima: 16 KB (non consigliato)</li><li>• -2: dimensione massima: 8 KB (consigliato)</li><li>• -1: dimensione massima: 4 KB (consigliato)</li><li>• I numeri positivi indicano l'archiviazione esattamente fino al numero specificato di elementi per nodo elenco.</li></ul>

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"> <li>• 0: disabilita completamente la compressione.</li> <li>• 1: la compressione inizia con il primo nodo successivo a head e termina con il primo nodo precedente a tail.  [head]-&gt;nodo-&gt;nodo-&gt;...-&gt;nodo-&gt;[tail]  Vengono compressi tutti i nodi tranne [head] e [tail].</li> <li>• 2: la compressione inizia con il secondo nodo successivo a head e termina con il secondo nodo precedente a tail.  [head]-&gt;[succ]-&gt;nodo-&gt;nodo-&gt;...-&gt;nodo-&gt;[prec]-&gt;[tail]  [head], [succ], [prec], [tail] non vengono compressi. Vengono compressi tutti gli altri nodi.</li> <li>• ecc.</li> </ul>

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 Valkey o Redis OSS (modalità cluster abilitata) in modalità cluster (sì) o di un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) in modalità non cluster (no). I gruppi di replica Valkey o Redis OSS (modalità cluster abilitata) in modalità cluster possono partizionare i propri dati su un massimo di 500 gruppi di nodi.</p> <p>* Redis 3.2. OSS x ha due gruppi di parametri predefiniti.</p> <ul style="list-style-type: none"><li>• <code>default.redis3.2</code> – Valore di default no.</li><li>• <code>default.redis3.2.cluster.on</code> – Valore di default yes.</li></ul> <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 Valkey o Redis OSS (modalità cluster abilitata) in modalità cluster smettono di accettare le query se rilevano che è stato scoperto almeno uno slot hash (nessun nodo disponibile lo serve). 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>
<code>hll-sparse-max-bytes</code>	<p>Impostazione predefinita: 3000</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: immediatamente</p>	<p>HyperLogLog limite di byte di rappresentazione sparsa. Il limite include l'intestazione a 16 byte. Quando si HyperLogLog utilizza la rappresentazione sparsa supera questo limite, viene convertita nella rappresentazione densa.</p> <p>Un valore maggiore di 16000 non è consigliato, perché a quel punto la rappresentazione densa è più efficiente in termini di memoria.</p> <p>Consigliamo un valore di circa 3000 per sfruttare i vantaggi della codifica efficiente in termini di spazio senza rallentamenti PFADD eccessivi, ossia <math>O(N)</math> con la codifica sparsa. Il valore può essere aumentato a ~10000 quando non CPU è un problema, ma lo spazio sì, e il set di dati è composto da molti HyperLogLogs con cardinalità compresa tra 0 e 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'impronta OSS dei dati di Redis aumenta fino a consumare 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. Riservando la memoria, puoi mettere da parte della memoria disponibile per OSS scopi diversi da Redis per ridurre la quantità di paging.</p> <p>Questo parametro è specifico e non fa parte della distribuzione Redis standard. ElastiCache OSS</p> <p>Per ulteriori informazioni, consulta <code>reserved-memory</code> e <a href="#">Gestione della memoria riservata per Valkey e Redis OSS</a>.</p>

### Parametri modificati in Redis OSS 3.2.4 (migliorato)

Famiglia del gruppo di parametri: `redis3.2`

Per Redis OSS 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	Non era modificabile

Nome	Informazioni	Modifica
	a un cluster di cache. In caso contrario, no.	
appendonly	Impostazione predefinita: off Modificabile: no	Se desideri eseguire l'aggiornamento da una OSS versione precedente di Redis, devi prima disattivarla. <code>appendonly</code>
appendfsync	Impostazione predefinita: off Modificabile: no	Se desideri eseguire l'aggiornamento da una OSS versione precedente di Redis, devi prima <code>appendfsync</code> disattivarla.
repl-timeout	Impostazione predefinita: 60 Modificabile: no	Ora non è modificabile con un valore di default di 60.
tcp-keepalive	Impostazione predefinita: 300	Il valore di default era 0.
list-max-ziplist-entries		Il parametro non è più disponibile.
list-max-ziplist-value		Il parametro non è più disponibile.

Parametri aggiunti per Redis OSS 2.8.24 (migliorato)

Famiglia del gruppo di parametri: `redis2.8`

Per Redis OSS 2.8.24 non sono supportati parametri aggiuntivi.

Parametri aggiunti per Redis OSS 2.8.23 (migliorato)

Famiglia del gruppo di parametri: `redis2.8`

Per Redis OSS 2.8.23 è supportato il seguente parametro aggiuntivo.

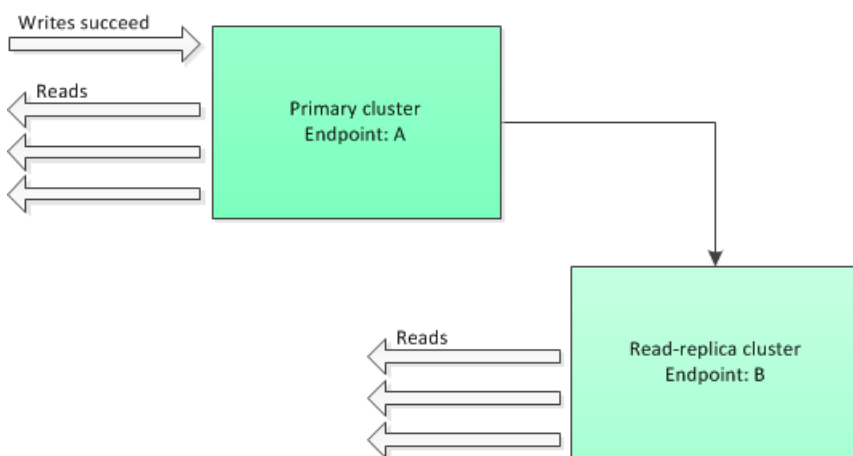


Nome	Informazioni	Descrizione
<code>close-on-slave-write</code>	Impostazione predefinita: <code>yes</code> (sì) Tipo: stringa ( <code>yes/no</code> ) Modificabile: sì Le modifiche diventano effettive: immediatamente	Se abilitato, i client che tentano di scrivere in una replica di sola lettura verranno disconnessi.

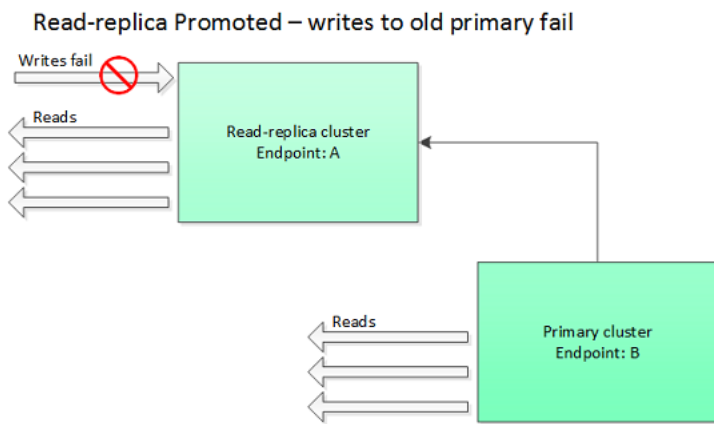
### Come funziona `close-on-slave-write`

Il `close-on-slave-write` parametro è stato introdotto da Amazon ElastiCache per darti un maggiore controllo sulla risposta del cluster quando un nodo primario e un nodo di replica di lettura scambiano i ruoli, promuovendo una replica di lettura a primaria.

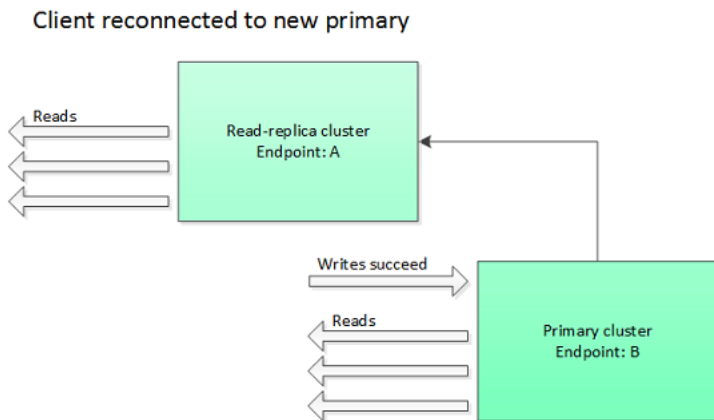
#### Before read-replica promotion



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. Questo è il comportamento di Redis OSS prima dell'ElastiCache introduzione `close-on-replica-write` e il comportamento se lo disabiliti `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 DNS tabella e riconnettersi all'endpoint primario, che ora sarebbe 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 Redis OSS 2.8.22 (migliorati)

Famiglia del gruppo di parametri: redis2.8

Per Redis OSS 2.8.22 non sono supportati parametri aggiuntivi.

 Important

- A partire dalla OSS versione 2.8.22 di Redis, `repl-backlog-size` si applica al cluster primario e ai cluster di replica.
- A partire dalla OSS versione Redis 2.8.22, il parametro non è supportato. `repl-timeout` Se viene modificato, ElastiCache verrà sovrascritto con il valore predefinito (60s), come facciamo con. `appendonly`

I parametri seguenti non sono più supportati.

- `appendonly`
- `appendfsync`
- `repl-timeout`

Parametri aggiunti da Redis OSS 2.8.21

Famiglia del gruppo di parametri: redis2.8

Per Redis OSS 2.8.21, non sono supportati parametri aggiuntivi.

Parametri aggiunti Redis 2.8.19 OSS

Famiglia del gruppo di parametri: redis2.8

Per Redis OSS 2.8.19 non sono supportati parametri aggiuntivi.

Parametri aggiunti da Redis 2.8.6 OSS


Famiglia del gruppo di parametri: redis2.8

Per Redis OSS 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 di <code>min-slaves-to-write</code>, a quel punto il primario smetterà di accettare scritture.</p> <p>Se questo parametro o <code>min-slaves-to-write</code> è 0, il nodo primario 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 primario 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 che Redis OSS può notificare ai clienti. Ogni tipo di evento è rappresentato da una lettera singola:</p> <ul style="list-style-type: none"> <li>• K— Eventi Keyspace, pubblicati con un prefisso di __keyspace@ &lt;db&gt; __</li> <li>• E— Eventi chiave-evento, pubblicati con un prefisso di __keyevent@ &lt;db&gt; __</li> <li>• g — Comandi generici e non specifici come DEL, EXPIRE, ecc. RENAME</li> <li>• \$— Comandi Stringa</li> <li>• I —Elencare comandi</li> <li>• s— Impostare i comandi</li> <li>• H— Comandi hash</li> <li>• z— Comandi impostati ordinati</li> <li>• x— Eventi scaduti (eventi generati ogni volta che una chiave scade)</li> <li>• e— Eventi sfrattati (eventi generati quando una chiave viene sfrattata per maxmemory)</li> <li>•</li> </ul>

Nome	Informazioni	Descrizione
		<p>A— Un alias per <math>\\$shzxe</math></p> <p>Può verificarsi una combinazione di questi tipi di eventi. Ad esempio, AKE significa che Redis OSS 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 keyspace 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" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>A partire da Redis OSS 2.8.22, questo parametro si applica al cluster primario 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"> <li>• Trasferimento di dati in blocco durante la sincronizzazione, dalla prospettiva della replica di lettura</li> <li>• Timeout del nodo principale dalla prospettiva della replica</li> <li>• Timeout della replica dalla prospettiva del nodo principale</li> </ul>




## Parametri Redis 2.6.13 OSS

Famiglia del gruppo di parametri: redis2.6

Redis OSS 2.6.13 è stata la prima versione di Redis supportata da OSS ElastiCache. La tabella seguente mostra i parametri di Redis OSS 2.6.13 supportati. 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 rielaborata dieci volte al secondo; ogni operazione di rehash richiede 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 funzione append only file di Redis (). AOF acquisisce tutti i comandi Redis che modificano i dati nella cache e viene utilizzato per il ripristino in seguito a determinati errori dei nodi.</p> <p>Il valore predefinito è no, il che significa che AOF è disattivato. Imposta questo parametro su yes per abilitarlo AOF.</p> <p>Per ulteriori informazioni, consulta <a href="#">Limitazione dell'impatto degli errori</a>.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Append Only Files (AOF) non è supportato per i nodi cache.t1.micro e</p> </div>

Nome	Informazioni	Descrizione
		<p>cache.t2.*. Per nodi di questo tipo, il valore del parametro <code>appendonly</code> viene ignorato.</p> <p> <b>Note</b> Per i gruppi di replica Multi-AZ, non AOF è consentito.</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>yes</code>, controlla la frequenza con cui il buffer AOF di output viene scritto su disco:</p> <ul style="list-style-type: none"> <li>• <code>no</code>— Il buffer viene scaricato su disco in base alle esigenze.</li> <li>• <code>ognisec</code>— il buffer viene svuotato una volta al secondo. Questa è l'impostazione predefinita.</li> <li>• <code>Sempre</code>: il buffer viene svuotato ogni volta che i dati nel cluster vengono modificati.</li> <li>• <code>Appendfsync</code> non è supportato per le versioni 2.8.22 e successive.</li> </ul>
<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 Redis OSS publish/subscribe: 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 Redis OSS publish/subscribe: se il buffer di output di un client raggiunge il numero di byte specificato, il client verrà disconnesso, ma solo se questa condizione persiste. <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 Redis OSS publish/subscribe: se il buffer di output di un client rimane in <code>client-output-buffer-limit-pubsub-soft-limit</code> byte per più di questo numero di secondi, il client verrà disconnesso.
<code>client-output-buffer-limit-slave-hard-limit</code>	<p>Valore di default: per i valori, vedere <a href="#">Parametri specifici del tipo di nodo Redis OSS</a></p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	Per le repliche di OSS 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 <a href="#">Parametri specifici del tipo di nodo Redis OSS</a></p> <p>Tipo: Integer</p> <p>Modificabile: no</p>	Per le repliche di OSS 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. <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 OSS lettura Redis: se il buffer di output di un client rimane in <code>client-output-buffer-limit-slave-soft-limit</code> byte per un periodo superiore a questo numero di secondi, 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 massimo di esecuzione per uno script Lua, in millisecondi, prima che venga eseguita l'azione necessaria per interrompere lo script. ElastiCache</p> <p>Se <code>lua-time-limit</code> viene superato, tutti i OSS comandi Redis restituiranno un errore nel formato <code>____- BUSY</code>. Poiché questo stato può causare interferenze con molte OSS operazioni i Redis essenziali, ElastiCache emetterà prima un comando. <code>SCRIPTKILL</code> Se ciò non riesce, ElastiCache riavvierà forzatamente Redis. OSS</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>Il numero massimo di client che possono essere connessi alla volta.</p>

Nome	Informazioni	Descrizione
	t2.small di default: 20000 Tipo: Integer Modificabile: no	
	t2.micro di default: 20000 Tipo: Integer Modificabile: no	
	t4g.micro di default: 20000 Tipo: Integer Modificabile: no	
	t3.medium di default: 46000 Tipo: Integer Modificabile: no	
	t3.small di default: 46000 Tipo: Integer Modificabile: no	
	t3.micro di default: 20000 Tipo: Integer Modificabile: no	

Nome	Informazioni	Descrizione
<code>maxmemory-policy</code>	<p>Impostazione predefinita: <code>volatile-lru</code></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: <code>volatile-lru</code>   <code>allkeys-lru</code>   <code>volatile-random</code>   <code>allkeys-random</code>   <code>volatile-ttl</code>   <code>noeviction</code></p> <p>Per ulteriori informazioni, consulta <a href="#">Utilizzo di Valkey o Redis come cache. OSS LRU</a></p>
<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 least-recently-used (LRU) e time-to-live (TTL), questo parametro rappresenta la dimensione del campione di chiavi da controllare. Per impostazione predefinita, Redis OSS sceglie 3 chiavi e utilizza quella utilizzata meno di recente.</p>



Nome	Informazioni	Descrizione
<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, il OSS nodo Redis crescerà fino a consumare quello del <code>maxmemory</code> nodo (vedi). <a href="#">Parametri specifici del tipo di nodo Redis OSS</a> In tal caso, le prestazioni del nodo subiranno una riduzione a causa di un eccessivo scambio di pagine di memoria. Riservando la memoria, è possibile riservare parte della memoria disponibile per OSS scopi diversi da Redis per ridurre la quantità di paging.</p> <p>Questo parametro è specifico e non fa parte della distribuzione Redis standard. ElastiCache OSS</p> <p>Per ulteriori informazioni, consulta <code>reserved-memory-percent</code> e <a href="#">Gestione della memoria riservata per Valkey e Redis OSS</a>.</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>
<code>slave-allow-chaining</code>	<p>Impostazione predefinita: no</p> <p>Tipo: stringa</p> <p>Modificabile: no</p>	<p>Determina se una replica di lettura in Redis OSS può avere repliche di lettura proprie.</p>

Nome	Informazioni	Descrizione
<code>slowlog-log-slower-than</code>	Impostazione predefinita: 10000 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	Il tempo massimo di esecuzione, in microsecondi, per i comandi che devono essere registrati dalla funzione Redis Slow Log. OSS
<code>slowlog-max-len</code>	Impostazione predefinita: 128 Tipo: Integer Modificabile: sì Le modifiche diventano effettive: immediatamente	La lunghezza massima del Redis OSS Slow Log.
<code>tcp-keepalive</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.  <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> <b>Important</b></p><p>Alcuni aspetti di questo parametro sono stati modificati nella OSS versione 3.2.4 di Redis. Per informazioni, consulta <a href="#">Parametri modificati in Redis OSS 3.2.4 (migliorato)</a>.</p></div>

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"> <li>• 0 - Non viene mai eseguita la disconnessione di un client inattivo.</li> <li>• 1-19 - Valori non validi.</li> <li>• &gt;=20 - Il numero di secondi di attesa da parte di un nodo prima della disconnessione di un client inattivo.</li> </ul>
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 specificate un gruppo di parametri per il cluster Redis OSS 2.6.13, verrà utilizzato un gruppo di parametri predefinito (`default.redis2.6`). 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 Redis OSS

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	lient-output-buffer-limitC - slave-hard-limit	C lient-output-buffe r-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	lient-output-buffer-limitC - slave-hard-limit	C lient-output-buffe r-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	lient-output-buffer-limitC - slave-hard-limit	C lient-output-buffe r-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	lient-output-buffer-limitC - slave-hard-limit	C lient-output-buffe r-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

Per impostazione predefinita, tutti i tipi di istanze di generazione corrente vengono creati in un Amazon Virtual VPC Private Cloud.

Le istanze T1 non supportano la funzione Multi-AZ.

Le istanze T1 e T2 non supportano Redis. OSS AOF

Le variabili di OSS configurazione Redis `appendonly` e `appendfsync` sono supportate nella versione OSS Redis 2.8.22 e successive.

## Parametri Memcached specifici

### Memcached

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 ElastiCache 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
<code>no_modern</code>	Impostazione predefinita: 1 Tipo: booleano Modificabile: sì Valori consentiti: 0,1 Modificazioni hanno effetto: All'avvio	Un alias per disabilitare i comandi <code>slab_reasign</code> , <code>lru_maintainer_thread</code> , <code>lru_segmented</code> e <code>maxconns_fast</code> Quando si utilizza Memcached 1.5 e versioni successive, imposta

Nome	Informazioni	Descrizione
		<p>no_modern anche hash_algorithm su. jenkins</p> <p>Inoltre, quando si utilizza Memcached 1.5.10, è controllato dal parametro. inline_ascii_reponse parallelly Ciò significa che se no_modern è disabilitato allora è disabilitato. inline_ascii_reponse Dal motore Memcached 1.5.16 in poi il inline_ascii_response parametro non si applica più, quindi no_modern essere abilitato o disabilitato non ha alcun effetto su. inline_ascii_reponse</p> <p>Se no_modern è disabilitato, allora,, e deve slab_reassign essere abilitato. lru_maintainer_thread lru_segmented maxconns_fast WILL Poiché hash_algorithm i parametri slab_automove e non sono SWITCH parametri , la loro impostazione si basa sulle configurazioni del gruppo di parametri.</p> <p>Se si desidera disabilitare no_modern e ripristinare aremodern, è necessario configurare un gruppo di parametri</p>

Nome	Informazioni	Descrizione
		<p>personalizzato per disabilitare questo parametro e quindi riavviare il sistema per rendere effettive le modifiche.</p> <div data-bbox="1008 430 1507 1411" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></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 predefinito aggiornato verrà rilevato automaticamente dai nuovi ElastiCache utenti per ciascuna regione dopo il 20 agosto 2021. ElastiCache Gli utenti esistenti nelle regioni prima del 20 agosto 2021 devono modificare manualmente i propri gruppi di parametri personalizzati per rilevare questa nuova modifica.</p></div>

Nome	Informazioni	Descrizione
<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. Piccolo rallentamento per ASCIIget, faster set.

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 Tipo: booleano Modificabile: Sì (richiede il riavvio, se impostato su <code>no_modern</code> )	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.

Nome	Informazioni	Descrizione
	Valori consentiti: 0,1 Modificazioni hanno effetto: Al lancio	

#### 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> .
<code>watch</code>	Impostazione predefinita: abilitato Tipo: booleano Modificabile: sì	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> .

Nome	Informazioni	Descrizione
	<p>Le modifiche diventano effettive: immediatamente</p> <p>I log possono essere eliminati se l'utente raggiunge i limiti relativi a <code>watcher_1ogbuf_size</code> e <code>worker_1ogbuf_size</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>
<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>

Nome	Informazioni	Descrizione
<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> <b>Note</b></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 1843"><p> <b>Note</b></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 modificare il valore del parametro . Per ulteriori informazioni, consulta <a href="#">Modifica di un gruppo di ElastiCache parametri</a>.</p></div>

Nome	Informazioni	Descrizione
<code>lru_maintainer</code>	<p>Impostazione predefinita: 0 (disabilitato)</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Le modifiche diventano effettive: Al lancio</p>	<p>Un thread in background che mescola gli elementi tra le LRU capacità raggiunte. Valori: 0, 1.</p>
<code>expirezero_does_not_evict</code>	<p>Impostazione predefinita: 0 (disabilitato)</p> <p>Tipo: booleano</p> <p>Modificabile: sì</p> <p>Modificazioni hanno effetto: Al lancio</p>	<p>Quando utilizzato con <code>lru_maintainer</code>, rende gli elementi non soggetti a espulsione con un tempo di scadenza pari a 0.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Warning</b></p> <p>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 ElastiCache hash, 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
s1ab_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 adotta un approccio lento e conservativo allo spostamento automatico delle lastre. Se è impostato su 2, sposta le lastre ElastiCache in modo aggressivo ogni volta che si verifica uno sfratto. Questa modalità è consigliabile esclusivamente a scopo di test.</p>

Nome	Descrizione
slab_reassign	Abilitare o disabilitare la riassegnazione dello slab. Se questo parametro è impostato su 1, puoi utilizzare il comando di "riassegnazione slab" (slab_reassign) 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
backlog_queue_limit	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 <a href="#">Modifica di un gruppo di ElastiCache parametri</a>.</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 check e set (CAS) saranno disabilitate e gli elementi archiviati consumeranno 8 byte in meno rispetto a quelli abilitati. CAS</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>	<p>Impostazione predefinita: 0(false)</p> <p>Tipo: Booleano</p> <p>Modificabile: sì</p> <p>Le modifiche hanno effetto: Dopo il riavvio</p>	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>	<p>Impostazione predefinita: 0(false)</p> <p>Tipo: Booleano</p> <p>Modificabile: no</p>	Se 1 (true), ElastiCache proverà a utilizzare pagine di memoria di grandi dimensioni.
<code>lock_down_paged_memory</code>	<p>Impostazione predefinita: 0(false)</p> <p>Tipo: Booleano</p> <p>Modificabile: no</p>	Se 1 (true), ElastiCache bloccherà tutta la memoria paginata.
<code>max_item_size</code>	<p>Impostazione predefinita: 1048576</p> <p>Tipo: Integer</p> <p>Modificabile: sì</p> <p>Le modifiche hanno effetto: Dopo il riavvio</p>	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 massimizzerà il limite del file principale.
<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 <a href="#">Sovraccarico delle connessioni Memcached</a> .
<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 predefinito, ElastiCache imposterà il valore su 13 `memcached_connections_overhead` 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 un Amazon Virtual Private Cloud (AmazonVPC).

## Ridimensionamento ElastiCache

Puoi scalare la ElastiCache cache in base alle tue esigenze. Le cache serverless e i cluster progettati internamente offrono diverse opzioni di scalabilità.

### ElastiCache Scalabilità 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 `()` per l'utilizzo dell'elaborazione. `ElastiCacheProcessingUnits` ECPU

## Impostazione dei limiti di dimensionamento per gestire i costi

È possibile scegliere di configurare un utilizzo massimo sia per l'archiviazione dei dati della cache sia per ECPU /second per la cache 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 valore di scalabilità massimo, l'applicazione potrebbe subire una riduzione delle prestazioni della cache quando la cache raggiunge il massimo. Quando impostate un valore massimo di archiviazione dei dati nella cache e l'archiviazione dei dati della cache raggiunge il massimo, ElastiCache inizierà a eliminare i dati dalla cache che hanno un set Time-To-Live (TTL), utilizzando la logica LRU. Se non ci sono dati che possono essere eliminati, le richieste di scrittura di dati aggiuntivi riceveranno un messaggio di errore Out Of Memory (OOM). Quando imposti un valore massimo di ECPU /second e l'utilizzo di calcolo del carico di lavoro supera questo valore, inizieranno a limitare le richieste. 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 preridimensionamento, 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 (ECPUs) 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 prescalatura utilizzando la console, oppure. ElastiCache CLI API ElastiCache Serverless aggiorna i ECPUs /second disponibili 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 ECPUs /secondo o l'archiviazione dei dati viene aggiornato tramite la console, oppure CLI/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 funzionalità Read from Replica. ElastiCache può raddoppiare /secondo ogni 10-12 minuti. ECPUs Questa velocità di scalabilità è sufficiente per la maggior parte dei carichi di lavoro. Se prevedi che un evento di scalabilità imminente possa superare questa frequenza, ti consigliamo di impostare il valore minimo ECPUs /secondo sul picco 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 valore minimo ECPUs al secondo o il nuovo spazio di archiviazione minimo. Ciò si verifica anche se l'applicazione non esegue le 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. ECPUs all'utilizzo 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 ECPUs €/secondo, ti verranno addebitati almeno 1,224 USD all'ora (utilizzando i prezzi ECPU in us-east-1), anche se l'utilizzo è inferiore al minimo impostato.
- ElastiCache Serverless supporta la scala minima richiesta a livello aggregato sulla cache. ElastiCache Serverless supporta anche un massimo di 30.000 al ECPUs secondo per slot (90.000 al ECPUs secondo quando si utilizza Read from Replica utilizzando connessioni). READONLY Come best practice, l'applicazione dovrebbe garantire che la distribuzione delle chiavi tra gli OSS slot Valkey o 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/>.
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 scala utilizzando il CLI, utilizzare il `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}'
```

## Rimuovere i limiti di scala utilizzando il CLI

Per rimuovere i limiti di scala utilizzando il CLI, impostare i parametri del 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}'
```

```
--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 dei cluster progettati autonomamente

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.

### Note

Se un OSS cluster Valkey o Redis viene replicato in una o più regioni, tali regioni vengono ridimensionate in ordine. Quando si esegue la scalabilità, vengono scalate prima le regioni secondarie e poi la regione principale. Quando si esegue la scala verso il basso, viene prima la regione principale e poi tutte le regioni secondarie.

Quando si aggiorna la versione del motore, l'ordine è Regione secondaria e quindi Regione principale.

### Argomenti

- [Scalabilità dei cluster per Memcached](#)
- [Scalabilità dei cluster per Valkey o Redis OSS \(modalità cluster disabilitata\)](#)
- [Scalabilità dei nodi di replica per Valkey o Redis OSS \(modalità cluster disabilitata\)](#)
- [Scalabilità dei cluster in Valkey o Redis OSS \(modalità cluster abilitata\)](#)

## Scalabilità dei cluster per Memcached

Le tabelle seguenti possono aiutarti a trovare l'argomento corretto per le azioni di ridimensionamento che desideri eseguire.

## Scalabilità dei cluster Memcached

I cluster Memcached sono composti da 1 a 60 nodi. Il dimensionamento orizzontale e verticale di un cluster Memcached è semplice come l'aggiunta al o la rimozione di nodi dal cluster.

Poiché è possibile ripartire i dati su tutti i nodi in un cluster Memcached, il dimensionamento verso l'alto a un tipo di nodo con una memoria di dimensioni maggiori è richiesto raramente. Tuttavia, poiché il motore Memcached non rende i dati permanenti, se si effettua il dimensionamento a un tipo di nodo differente, il nuovo cluster è inizialmente vuoto a meno che non venga popolato dall'applicazione.

### Dimensionamento dei cluster Memcached

Azione	Argomento
Dimensionamento orizzontale	<a href="#">Aggiunta di nodi a un cluster</a>
Dimensionamento verticale	<a href="#">Eliminazione di nodi da un cluster</a>
Modifica dei tipi di nodo	<a href="#">Dimensionamento verticale di Memcached</a>

I cluster Memcached sono composti da 1 a 60 nodi. Il dimensionamento orizzontale e verticale di un cluster Memcached è semplice come l'aggiunta al o la rimozione di nodi dal cluster.

Poiché è possibile ripartire i dati su tutti i nodi in un cluster Memcached, il dimensionamento verso l'alto a un tipo di nodo con una memoria di dimensioni maggiori è richiesto raramente. Tuttavia, poiché il motore Memcached non rende i dati permanenti, se si effettua il dimensionamento a un tipo di nodo differente, il nuovo cluster è inizialmente vuoto a meno che non venga popolato dall'applicazione.

### Argomenti

- [Dimensionamento orizzontale dei cluster Memcached](#)
- [Dimensionamento verticale di Memcached](#)

### Dimensionamento orizzontale dei cluster Memcached

Il motore Memcached supporta il partizionamento dei dati su più nodi. Per questo motivo, il dimensionamento orizzontale dei cluster Memcached è più semplice. Un cluster Memcached

può avere da 1 a 60 nodi. Per dimensionare orizzontalmente un cluster Memcached è sufficiente aggiungere o rimuovere i nodi.

Gli argomenti che seguono illustrano nel dettaglio come dimensionare un cluster Memcached verso l'alto o verso il basso aggiungendo o rimuovendo nodi.

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

Ogni volta che si cambia il numero di nodi di un cluster Memcached, è necessario rimappare almeno una parte del keyspace in modo da mapparlo sul nodo corretto. Per informazioni più dettagliate sul load balancer del cluster Memcached, consulta [Configurazione ElastiCache del client per un bilanciamento efficiente del carico \(Memcached\)](#):

Se utilizzi il rilevamento automatico sul cluster Memcached, non è necessario modificare gli endpoint nell'applicazione quando aggiungi o rimuovi nodi. Per ulteriori informazioni sull'individuazione automatica, consulta [Identifica automaticamente i nodi del cluster \(Memcached\)](#). Se non utilizzi l'individuazione automatica, ogni volta che modifichi il numero di nodi nel cluster Memcached è necessario aggiornare gli endpoint nell'applicazione.

### Dimensionamento verticale di Memcached

Quando effettui il dimensionamento verso l'alto o verso il basso di un cluster Memcached, devi creare un nuovo cluster. Al momento della creazione, i cluster Memcached sono sempre vuoti, a meno che l'applicazione non li popoli.

#### Important

Se esegui il dimensionamento a un tipo di nodo più piccolo, assicurati che il tipo di nodo più piccolo sia adeguato ai dati e all'overhead. Per ulteriori informazioni, consulta [Select cache node size](#).

### Argomenti

- [Dimensionamento verticale di Memcached \(console\)](#)
- [Dimensionamento verticale di Memcached \(AWS CLI\)](#)
- [Dimensionamento verticale di Memcached \(ElastiCache API\)](#)

## Dimensionamento verticale di Memcached (console)

La procedura seguente illustra come scalare verticalmente il cluster utilizzando la console. ElastiCache

Per dimensionare verticalmente un cluster Memcached (console)

1. Creare un nuovo cluster con il nuovo tipo di nodo. Per ulteriori informazioni, consulta [Creazione di un cluster Memcached \(console\)](#).
2. Nell'applicazione, aggiornare gli endpoint con gli endpoint del nuovo cluster. Per ulteriori informazioni, consulta [Ricerca degli endpoint di un cluster \(console\) \(Memcached\)](#).
3. Eliminare il cluster precedente. Per ulteriori informazioni, consulta [Deleting a new node in Memcached](#).

## Dimensionamento verticale di Memcached (AWS CLI)

La procedura seguente descrive le diverse fasi per eseguire il dimensionamento verticale del cluster di cache Memcached tramite AWS CLI.

Per dimensionare verticalmente un cluster di cache Memcached (AWS CLI)

1. Creare un nuovo cluster di cache con il nuovo tipo di nodo. Per ulteriori informazioni, vedere [Creazione di cluster](#) con. CLI
2. Nell'applicazione, aggiornare gli endpoint con gli endpoint del nuovo cluster. Per ulteriori informazioni, consulta [Individuazione degli endpoint \(AWS CLI\)](#).
3. Eliminare il cluster di cache precedente. Per ulteriori informazioni, consulta [Utilizzo di AWS CLI per eliminare un ElastiCache cluster](#).

## Dimensionamento verticale di Memcached (ElastiCache API)

La procedura seguente illustra come scalare verticalmente il cluster di cache Memcached utilizzando. ElastiCache API

Per scalare verticalmente un cluster di cache Memcached () ElastiCache API

1. Creare un nuovo cluster di cache con il nuovo tipo di nodo. Per ulteriori informazioni, consulta [Creazione di un cluster per Memcached \(\) ElastiCache API](#)

2. Nell'applicazione, aggiornare gli endpoint sugli endpoint del nuovo cluster di cache. Per ulteriori informazioni, consulta [Ricerca degli endpoint \(\) ElastiCache API](#).
3. Eliminare il cluster di cache precedente. Per ulteriori informazioni, consulta [Utilizzo del ElastiCache API](#).



## Scalabilità dei cluster per Valkey o Redis OSS (modalità cluster disabilitata)

I cluster Valkey o Redis OSS (modalità cluster disabilitata) possono essere cluster a nodo singolo con 0 shard o cluster multinodo 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.

### Argomenti

- [Scalabilità di cluster a nodo singolo per Valkey o Redis \(modalità cluster disabilitata\) OSS](#)

### OSS Scalabilità dei cluster Valkey o Redis

Azione	Valkey o Redis OSS (modalità cluster disabilitata)	Valkey o Redis OSS (modalità cluster abilitata)
Dimensionamento verticale	<a href="#">Rimozione di nodi da un ElastiCache cluster</a>	<a href="#">Scalabilità dei cluster in Valkey o Redis OSS (modalità cluster abilitata)</a>
Dimensionamento orizzontale	<a href="#">Aggiunta di nodi a un cluster</a>	<a href="#">Resharding online per Valkey o Redis OSS (modalità cluster abilitata)</a>
Modifica dei tipi di nodo	<p>Verso un tipo di nodo più grande:</p> <ul style="list-style-type: none"> <li>• <a href="#">Scalabilità di cluster Valkey o Redis a nodo singolo OSS</a></li> <li>• <a href="#">Scalabilità dei cluster Valkey o Redis con repliche OSS</a></li> </ul> <p>Verso un tipo di nodo più piccolo:</p> <ul style="list-style-type: none"> <li>•</li> </ul>	<a href="#">Ridimensionamento verticale online tramite la modifica del tipo di nodo</a>

Azione	Valkey o Redis OSS (modalità cluster disabilitata)	Valkey o Redis OSS (modalità cluster abilitata)
	<ul style="list-style-type: none"> <li><a href="#">Ridimensionamento dei cluster Valkey o Redis a nodo singolo OSS</a></li> <li>• <a href="#">Ridimensionamento dei cluster Valkey o Redis OSS con repliche</a></li> </ul>	
Modifica del numero di gruppi del nodo	Non supportato per i cluster Valkey o Redis OSS (modalità cluster disabilitata)	<a href="#">Scalabilità dei cluster in Valkey o Redis OSS (modalità cluster abilitata)</a>

## Indice

- [Scalabilità di cluster a nodo singolo per Valkey o Redis \(modalità cluster disabilitata\) OSS](#)
  - [Scalabilità di cluster Valkey o Redis a nodo singolo OSS](#)
    - [Scalabilità di cluster a nodo singolo per Valkey o Redis OSS \(modalità cluster disabilitata\) \(console\)](#)
    - [Scalabilità dei cluster di cache Valkey o Redis a nodo singolo \(\) OSS AWS CLI](#)
    - [Scalabilità dei cluster di cache Valkey o Redis OSS a nodo singolo \(\) ElastiCache API](#)
  - [Ridimensionamento dei cluster Valkey o Redis a nodo singolo OSS](#)
    - [Ridimensionamento di un cluster Valkey o Redis a nodo singolo \(console\) OSS](#)
    - [Ridimensionamento dei cluster di cache Valkey o Redis a nodo singolo \(\) OSS AWS CLI](#)
    - [Ridimensionamento dei cluster di cache Valkey o Redis OSS a nodo singolo \(\) ElastiCache API](#)

## Scalabilità di cluster a nodo singolo per Valkey o Redis (modalità cluster disabilitata) OSS

I nodi Valkey o Redis OSS (modalità cluster disabilitata) devono essere sufficientemente grandi da contenere tutti i dati della cache più l'overhead di Valkey o Redis. OSS Per modificare la capacità di dati del cluster Valkey o Redis OSS (modalità cluster disabilitata), è necessario scalare verticalmente,

passando a un tipo di nodo più grande per aumentare la capacità dei dati o ridimensionarlo a un tipo di nodo più piccolo per ridurre la capacità dei dati.

Il processo ElastiCache di scalabilità è progettato per fare il massimo sforzo per conservare i dati esistenti e richiede una replica Valkey o Redis di successo. OSS Per i cluster Valkey o Redis OSS (modalità cluster disabilitata), si consiglia di rendere disponibile una memoria sufficiente per Valkey o Redis. OSS

Non è possibile partizionare i dati su più cluster Valkey o Redis OSS (modalità cluster disabilitata). Tuttavia, se è necessario solo aumentare o diminuire la capacità di lettura del cluster, è possibile creare un cluster Valkey o Redis OSS (modalità cluster disabilitata) con nodi di replica e aggiungere o rimuovere repliche di lettura. Per creare un cluster Valkey o Redis OSS (modalità cluster disabilitata) con nodi di replica utilizzando il cluster di cache Valkey o OSS Redis a nodo singolo come cluster principale, consulta [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)

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, consulta [Aumento della capacità di lettura](#) o [Diminuzione della capacità di lettura](#).

Oltre a essere in grado di scalare la capacità di lettura, i cluster Valkey o Redis OSS (modalità cluster disabilitata) con repliche offrono altri vantaggi aziendali. Per ulteriori informazioni, consulta [Alta disponibilità utilizzando gruppi di replica](#).

#### Important

Se il tuo gruppo di parametri utilizza `reserved-memory` la memoria per il OSS sovraccarico di Valkey o Redis, prima di iniziare il ridimensionamento assicurati di disporre di un gruppo di parametri personalizzato che riservi la quantità di memoria corretta 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 riservata per Valkey e Redis OSS](#).

## Argomenti

- [Scalabilità di cluster Valkey o Redis a nodo singolo OSS](#)
- [Ridimensionamento dei cluster Valkey o Redis a nodo singolo OSS](#)



## Scalabilità di cluster Valkey o Redis a nodo singolo OSS

Quando si esegue la scalabilità di un OSS cluster Valkey o Redis a nodo singolo, ElastiCache esegue il seguente processo, indipendentemente dal fatto che si utilizzi la console, o la. ElastiCache AWS CLI ElastiCache API

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. Durante l'aggiornamento della voce, noterai una breve interruzione (alcuni secondi) delle letture e delle scritture dal nodo principale. DNS
4. ElastiCache elimina il vecchio cluster di cache. 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 OSS scale-up di Valkey o Redis è bloccata se è previsto un aggiornamento del motore per la prossima finestra di manutenzione. Per ulteriori informazioni sulla manutenzione Windows, consultare [Gestione della manutenzione dei ElastiCache cluster](#).

### Operazioni Valkey o Redis bloccate OSS

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.

- Pianifica l'operazione di OSS scalabilità di Valkey o Redis per la prossima finestra di manutenzione deselezionando la casella di controllo Applica immediatamente (CLLuse:, use:). `--no-apply-immediately` API `ApplyImmediately=false`
- Attendi la finestra di manutenzione successiva (o successiva) per eseguire l'operazione di scalabilità con Valkey o Redis. OSS
- Aggiungi l'aggiornamento del OSS motore Valkey o Redis a questa modifica del cluster di cache selezionando la casella di controllo Applica immediatamente (CLLuse:--apply-immediately, API use:). `ApplyImmediately=true` In questo modo si sblocca l'operazione di dimensionamento verso l'alto per eseguire immediatamente l'aggiornamento automatico del motore.

È possibile scalare un cluster Valkey o Redis a nodo singolo OSS (modalità cluster disabilitata) utilizzando la ElastiCache console, il, o. AWS CLI ElastiCache API

#### Important

Se il gruppo di parametri utilizza lo `reserved-memory` scopo di riservare memoria per il OSS sovraccarico di Valkey o Redis, prima di iniziare la scalabilità assicurati di disporre di un gruppo di parametri personalizzato che riservi la quantità di memoria corretta 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 riservata per Valkey e Redis OSS](#).

Scalabilità di cluster a nodo singolo per Valkey o Redis OSS (modalità cluster disabilitata) (console)

La procedura seguente descrive come scalare un cluster Valkey o Redis a nodo singolo utilizzando la console di gestione. OSS ElastiCache Durante questo processo, il OSS cluster Valkey o Redis continuerà a soddisfare le richieste con tempi di inattività minimi.

## Per scalare un cluster Valkey o Redis a nodo singolo (console) OSS

1. Accedi a AWS Management Console e apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o Redis OSS.
3. Dall'elenco dei cluster, scegli il cluster che desideri scalare (deve eseguire il motore Valkey o Redis, non il OSS motore Valkey o Redis in cluster). OSS
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.

## Scalabilità dei cluster di cache Valkey o Redis a nodo singolo () OSS AWS CLI

La procedura seguente descrive come scalare un cluster di cache Valkey o Redis a nodo singolo utilizzando. OSS AWS CLI Durante questo processo, il OSS cluster Valkey o Redis continuerà a soddisfare le richieste con tempi di inattività minimi.

## Per scalare un cluster di cache Valkey o Redis a nodo singolo () OSS AWS CLI

1. Determina i tipi di nodi fino a cui puoi scalare eseguendo il AWS CLI `list-allowed-node-type-modifications` comando 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 precedente ha un aspetto simile JSON al seguente (formato).

```
{
 "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 la pagina [list-allowed-node-type-modifiche](#) nel riferimento.AWS CLI



2. Modifica il cluster di cache esistente specificando il cluster di cache da scalare verso l'alto e il nuovo tipo di nodo più grande, utilizzando il AWS CLI `modify-cache-cluster` comando 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 precedente ha un aspetto simile al seguente (JSONformato).

```
{
 "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 la pagina [modify-cache-cluster](#) nel AWS CLI riferimento.

3. Se hai usato il `--apply-immediately`, controlla lo stato del nuovo cluster di cache usando il AWS CLI `describe-cache-clusters` comando con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più grande.
  - `--cache-cluster-id` `cluster-id`— Il nome del tuo cluster di cache Valkey o Redis OSS 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 la pagina [describe-cache-clusters](#) nel Reference.AWS CLI

## Scalabilità dei cluster di cache Valkey o Redis OSS a nodo singolo () ElastiCache API

La procedura seguente descrive come scalare un cluster di cache Valkey o Redis a nodo singolo utilizzando l'API ElastiCache OSS. Durante questo processo, il cluster Valkey o Redis continuerà a soddisfare le richieste con tempi di inattività minimi.

Per scalare un cluster di cache Valkey o Redis a nodo singolo () OSS ElastiCache API

1. Determina i tipi di nodi fino a cui puoi scalare eseguendo l'azione `ListAllowedNodeTypeModifications` ElastiCache API con il seguente parametro.
  - `CacheClusterId`— Il nome del cluster di OSS cache Valkey o Redis a nodo singolo che desideri scalare.

```
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 la pagina [ListAllowedNodeTypeModifications](#) nell'Amazon ElastiCache API Reference.

2. Modifica il tuo cluster di cache esistente specificando il cluster di cache da scalare verso l'alto e il nuovo tipo di nodo più grande, utilizzando l'azione `ModifyCacheCluster` ElastiCache API 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'azione `ListAllowedNodeTypeModifications` nel passaggio precedente.
  - `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 la pagina [ModifyCacheCluster](#) nell'Amazon ElastiCache API Reference.

3. Se lo hai utilizzato `ApplyImmediately=true`, controlla lo stato del nuovo cluster di cache utilizzando l' `ElastiCache APIDescribeCacheClusters` azione 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 tuo cluster di cache Valkey o Redis OSS 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 la pagina [DescribeCacheClusters](#) nell'Amazon ElastiCache API Reference.



## Ridimensionamento dei cluster Valkey o Redis a nodo singolo OSS

Le seguenti sezioni illustrano come ridimensionare un cluster Valkey o Redis OSS a nodo singolo fino a un tipo di nodo più piccolo. Garantire che il nuovo tipo di nodo più piccolo sia sufficientemente grande da contenere tutti i dati e il OSS sovraccarico di Valkey o Redis è importante per il successo a lungo termine del nuovo cluster Valkey o Redis. OSS Per ulteriori informazioni, consulta [Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS](#).

### 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

- [Ridimensionamento di un cluster Valkey o Redis a nodo singolo \(console\) OSS](#)
- [Ridimensionamento dei cluster di cache Valkey o Redis a nodo singolo \(\) OSS AWS CLI](#)
- [Ridimensionamento dei cluster di cache Valkey o Redis OSS a nodo singolo \(\) ElastiCache API](#)

### Ridimensionamento di un cluster Valkey o Redis a nodo singolo (console) OSS

La procedura seguente illustra come ridimensionare il OSS cluster Valkey o Redis a nodo singolo fino a un tipo di nodo più piccolo utilizzando la console. ElastiCache

### Important

Se il tuo gruppo di parametri utilizza `reserved-memory` la memoria per il OSS sovraccarico di Valkey o Redis, prima di iniziare il ridimensionamento assicurati di disporre di un gruppo di parametri personalizzato che riservi la quantità di memoria corretta 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 riservata per Valkey e Redis OSS](#).

## Per ridimensionare il cluster Valkey o Redis a nodo singolo (console) OSS

1. Verificare che il dimensionamento a un tipo di nodo più piccolo sia adeguato ai dati e all'overhead.
2. Se il tuo gruppo di parametri utilizza `reserved-memory` la memoria per il OSS sovraccarico di Valkey o Redis, assicurati di disporre di un gruppo di parametri personalizzato per mettere da parte la quantità di memoria corretta 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 riservata per Valkey e Redis OSS](#).

3. Accedi a AWS Management Console e apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
4. Nell'elenco dei cluster, scegliere quello da diminuire. Questo cluster deve utilizzare il motore Valkey o Redis e non il OSS motore Valkey o Redis in cluster. OSS
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.

## Ridimensionamento dei cluster di cache Valkey o Redis a nodo singolo () OSS AWS CLI

La procedura seguente descrive come ridimensionare un cluster di cache Valkey o Redis a nodo singolo utilizzando. OSS AWS CLI

## Per ridimensionare un cluster di cache Valkey o Redis a nodo singolo () OSS AWS CLI

1. Determina i tipi di nodi a cui puoi ridimensionare eseguendo il AWS CLI `list-allowed-node-type-modifications` comando 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 precedente ha un aspetto simile JSON al seguente (formato).

```
{
 "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 la pagina [list-allowed-node-type-modifiche](#) nel riferimento.AWS CLI

2. Modifica il cluster di cache esistente specificando il cluster di cache da ridimensionare e il nuovo tipo di nodo più piccolo, utilizzando il AWS CLI `modify-cache-cluster` comando 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 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-xl \
 --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-xl ^
 --apply-immediately
```

L'output del comando precedente ha un aspetto simile al seguente (JSONformato).

```
{
 "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 la pagina [modify-cache-cluster](#) nel AWS CLI riferimento.

3. Se hai usato il `--apply-immediately`, controlla lo stato del nuovo cluster di cache usando il AWS CLI `describe-cache-clusters` comando 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 tuo cluster di cache Valkey o Redis OSS 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 la pagina [describe-cache-clusters](#) nel Reference.AWS CLI

## Ridimensionamento dei cluster di cache Valkey o Redis OSS a nodo singolo () ElastiCache API

La procedura seguente descrive come scalare verso l'alto un cluster di cache Valkey o Redis a nodo singolo utilizzando. OSS ElastiCache API

Per ridimensionare un cluster di cache Valkey o Redis a nodo singolo () OSS ElastiCache API

1. Determina i tipi di nodi a cui puoi ridimensionare eseguendo l' ElastiCache API `ListAllowedNodeTypeModifications` azione con il seguente parametro.
  - `CacheClusterId`— Il nome del cluster di OSS cache Valkey o Redis a nodo singolo che desideri ridimensionare.

```
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 la pagina [ListAllowedNodeTypeModifications](#) nell'Amazon ElastiCache API Reference.

2. Modifica il tuo cluster di cache esistente specificando il cluster di cache da scalare verso l'alto e il nuovo tipo di nodo più grande, utilizzando l'`ModifyCacheCluster` ElastiCache API azione 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'`ListAllowedNodeTypeModifications` azione nel passaggio precedente.

- `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 la pagina [ModifyCacheCluster](#) nell'Amazon ElastiCache API Reference.

3. Se lo hai utilizzato `ApplyImmediately=true`, controlla lo stato del nuovo cluster di cache utilizzando l' `ElastiCache APIDescribeCacheClusters` azione 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 tuo cluster di cache Valkey o Redis OSS 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 la pagina [DescribeCacheClusters](#) nell'Amazon ElastiCache API Reference.

## Scalabilità dei nodi di replica per Valkey o Redis OSS (modalità cluster disabilitata)

Un OSS cluster Valkey o Redis con nodi di replica (chiamato gruppo di replica in API/CLI) offre un'elevata disponibilità tramite la replica con Multi-AZ con failover automatico abilitato. Un cluster con nodi di replica è una raccolta logica di un massimo di sei OSS nodi Valkey o Redis in cui un nodo, il primario, è in grado di soddisfare 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 Valkey o Redis OSS (modalità cluster disabilitata) non supportano il partizionamento dei dati su più cluster, ogni nodo di un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) contiene l'intero set di dati della cache. I cluster Valkey o Redis OSS (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 ElastiCache di scalabilità è progettato per fare il massimo sforzo per conservare i dati esistenti e richiede una replica efficace con Valkey o Redis. OSS Per i OSS cluster Valkey o Redis con repliche, si consiglia di rendere disponibile una memoria sufficiente per Valkey o Redis. OSS

### Argomenti

- [Scalabilità dei cluster Valkey o Redis con repliche OSS](#)
- [Ridimensionamento dei cluster Valkey o Redis OSS con repliche](#)
- [Aumento della capacità di lettura](#)
- [Diminuzione della capacità di lettura](#)

### Argomenti correlati

- [Alta disponibilità utilizzando gruppi di replica](#)
- [Replica: modalità OSS cluster Valkey e Redis disabilitata o abilitata](#)
- [Riduzione al minimo dei tempi di inattività ElastiCache utilizzando Multi-AZ con Valkey e Redis OSS](#)
- [Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS](#)

## Argomenti

- [Scalabilità dei cluster Valkey o Redis con repliche OSS](#)
- [Ridimensionamento dei cluster Valkey o Redis OSS con repliche](#)
- [Aumento della capacità di lettura](#)
- [Diminuzione della capacità di lettura](#)

## Scalabilità dei cluster Valkey o Redis con repliche OSS

Amazon ElastiCache fornisce console e API supporto per scalare il gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata). CLI

Quando viene avviato il processo di scalabilità verticale, effettua le seguenti operazioni: ElastiCache

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 DNS voci in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Valkey 7.2 e versioni successive o Redis OSS 5.0.5 e versioni successive, puoi scalare i cluster abilitati al failover automatico mentre il cluster continua a rimanere online e a soddisfare le richieste in arrivo. Nella OSS versione 4.0.10 e precedenti di Redis, potresti notare una breve interruzione delle operazioni di lettura e scrittura sulle versioni precedenti dal nodo primario durante l'aggiornamento della voce. DNS
5. Elimina i vecchi nodi (/: gruppo di replica)CLI. API 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 OSS scalabilità verticale di Valkey o Redis è bloccata se è pianificato un aggiornamento del motore per la prossima finestra di manutenzione del cluster.

### Operazioni Valkey o Redis bloccate OSS

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 sospenso che ti blocca, puoi svolgere una delle seguenti operazioni.



- Pianifica l'operazione di OSS scalabilità di Valkey o Redis per la prossima finestra di manutenzione deselezionando la casella di controllo Applica immediatamente (CLluse:, use:). `--no-apply-immediately` API `ApplyImmediately=false`
- Attendi la finestra di manutenzione successiva (o successiva) per eseguire l'operazione di scale-up con Valkey o Redis. OSS
- Aggiungi l'aggiornamento del OSS motore Valkey o Redis a questa modifica del cluster di cache selezionando la casella di controllo Applica immediatamente (CLluse:, use:). `--apply-immediately` API `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 scalare il OSS cluster Valkey o Redis con repliche attive utilizzando la ElastiCache console, il e il. AWS CLI ElastiCache API

#### Important

Se il tuo gruppo di parametri serve `reserved-memory` a riservare memoria per il OSS sovraccarico di Valkey o Redis, prima di iniziare il ridimensionamento assicurati di disporre di un gruppo di parametri personalizzato che riservi la quantità di memoria corretta 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 riservata per Valkey e Redis OSS](#).

## Scalabilità di un cluster Valkey o Redis OSS 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.

Il seguente processo ridimensiona il cluster con repliche dal tipo di nodo corrente a un nuovo tipo di nodo più grande utilizzando la console. ElastiCache Durante questo processo, potrebbe verificarsi una breve interruzione delle operazioni di lettura e scrittura per altre versioni dal nodo primario durante l'aggiornamento della DNS voce. È possibile che si verifichi un tempo di inattività inferiore a 1 secondo per i nodi che eseguono versioni 5.0.6 e successive e alcuni secondi per le versioni precedenti.

## Per scalare il cluster Valkey o Redis con repliche (console) OSS

1. Accedi a AWS Management Console e apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS
3. Nell'elenco di cluster, scegliere quello da dimensionare. Questo cluster deve eseguire il motore Valkey o Redis e non il OSS motore Valkey o Redis in cluster. OSS
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.

## Scalabilità di un gruppo di replica Valkey o Redis () OSS 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 aggiorna le DNS voci in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Valkey 7.2 e versioni successive o Redis OSS 5.0.5 e versioni successive, puoi scalare i cluster abilitati al failover automatico mentre il cluster continua a rimanere online e a soddisfare le richieste in arrivo. Nella versione 4.0.10 e successive, è possibile notare una breve interruzione delle operazioni di lettura e scrittura sulle versioni precedenti dal nodo principale durante l'aggiornamento della voce. DNS

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 scalare un Valkey o Redis Replication Group () OSS AWS CLI

1. Determina a quali tipi di nodi puoi scalare eseguendo il AWS CLI `list-allowed-node-type-modifications` comando 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 JSON al seguente (formato).

```
{
 "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 la pagina [list-allowed-node-type-modifiche](#) nel riferimento.AWS CLI

2. Ridimensiona il gruppo di replica corrente fino al nuovo tipo di nodo utilizzando il AWS CLI `modify-replication-group` comando 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 `list-allowed-node-type-modifications` comando nel passaggio precedente.
  - `--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 ha un aspetto simile JSON al seguente (formato).

```
{
 "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"
 }
 },
 {
 "CurrentRole": "replica",
 "PreferredAvailabilityZone": "us-west-2c",
 "CacheNodeId": "0001",
 "ReadEndpoint": {
 "Port": 6379,
 "Address": "my-repl-group-002.8fdx4s.0001.usw2.cache.amazonaws.com"
 }
 }
],
 "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 la pagina [modify-replication-group](#) nel AWS CLI riferimento.

3. Se avete utilizzato il `--apply-immediately` parametro, monitorate lo stato del gruppo di replica utilizzando il AWS CLI `describe-replication-group` comando con il seguente parametro. Sebbene lo stato sia ancora in fase di modifica, è possibile che si verifichi un tempo di inattività inferiore a 1 secondo per i nodi in esecuzione su versioni 5.0.6 e successive e una breve interruzione delle operazioni di lettura e scrittura per le versioni precedenti dal nodo primario durante l'aggiornamento della voce. DNS
- `--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 la sezione Reference. [describe-replication-groups](#) AWS CLI

## Scalabilità di un gruppo di OSS replica Valkey o Redis () ElastiCache API

Il processo seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo di nodo più grande utilizzando il. ElastiCache API Per Valkey 7.2 e versioni successive o Redis OSS 5.0.5 e versioni successive, puoi scalare i cluster abilitati al failover automatico mentre il cluster continua a rimanere online e a soddisfare le richieste in arrivo. Nella versione Redis OSS 4.0.10 e precedenti, è possibile notare una breve interruzione delle operazioni di lettura e scrittura sulle versioni precedenti dal nodo primario durante l'aggiornamento della voce. DNS

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 scalare un Valkey o un Redis Replication Group () OSS ElastiCache API

1. Determina a quali tipi di nodi puoi scalare utilizzando l' ElastiCache API `ListAllowedNodeTypeModifications` azione 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 la pagina [ListAllowedNodeTypeModifications](#) nell'Amazon ElastiCache API Reference.

2. Scala il tuo attuale gruppo di replica fino al nuovo tipo di nodo utilizzando l'`ModifyRedplicationGroup` ElastiCache API azione 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'`ListAllowedNodeTypeModifications` azione nel passaggio precedente.
- `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 la pagina [ModifyReplicationGroup](#) nell'Amazon ElastiCache API Reference.

3. Se lo hai utilizzato `ApplyImmediately=true`, monitora lo stato del gruppo di replica utilizzando l' `ElastiCache APIDescribeReplicationGroups` azione 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 la pagina [DescribeReplicationGroups](#) nell'Amazon ElastiCache API Reference.



## Ridimensionamento dei cluster Valkey o Redis OSS con repliche

Le sezioni seguenti illustrano come ridimensionare un cluster di cache Valkey o Redis OSS (modalità cluster disabilitata) con nodi di replica fino 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 [Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS](#).

### 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 tuo gruppo di parametri utilizza `reserved-memory` la memoria per l'OSS sovraccarico di Valkey o Redis, prima di iniziare il ridimensionamento assicurati di disporre di un gruppo di parametri personalizzato che riservi la giusta quantità 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 riservata per Valkey e Redis OSS](#).

## Ridimensionamento di un cluster Valkey o Redis OSS con repliche (console)

Il seguente processo ridimensiona il OSS cluster Valkey o Redis con nodi di replica su un tipo di nodo più piccolo utilizzando la console. ElastiCache

Per ridimensionare un OSS cluster Valkey o Redis con 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 tuo gruppo di parametri serve `reserved-memory` a riservare memoria per il OSS sovraccarico di Valkey o Redis, assicurati di disporre di un gruppo di parametri personalizzato per mettere da parte la quantità di memoria corretta 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 riservata per Valkey e Redis OSS](#).

3. Accedi a AWS Management Console e apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
4. Nell'elenco dei cluster, scegliere quello da diminuire. Questo cluster deve utilizzare il motore Valkey o Redis e non il OSS motore Valkey o Redis in cluster. OSS
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.

## Ridimensionamento di un gruppo di replica Valkey o Redis () OSS 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 aggiorna le DNS voci in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Valkey 7.2 o versioni successive o Redis OSS 5.0.5 e versioni successive, puoi scalare i cluster abilitati al failover automatico mentre il cluster continua a rimanere online e a soddisfare le richieste in arrivo. Nella versione 4.0.10 e precedenti, è possibile notare una breve interruzione delle operazioni di lettura e scrittura sulle versioni precedenti dal nodo principale durante l'aggiornamento della voce. DNS

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 ridimensionare un Valkey o Redis Replication Group () OSS AWS CLI

1. Determina a quali tipi di nodi puoi ridimensionare eseguendo il AWS CLI `list-allowed-node-type-modifications` comando 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 JSON al seguente (formato).

```
{
 "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 la pagina [list-allowed-node-type-modifiche](#) nel riferimento.AWS CLI

2. Ridimensiona il gruppo di replica corrente fino al nuovo tipo di nodo utilizzando il AWS CLI `modify-replication-group` comando 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 `list-allowed-node-type-modifications` comando nel passaggio precedente.
- `--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 ha un aspetto simile JSON al seguente (formato).

```
{"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 la pagina [modify-replication-group](#) nel AWS CLI riferimento.

3. Se avete utilizzato il `--apply-immediately` parametro, monitorate lo stato del gruppo di replica utilizzando il AWS CLI `describe-replication-group` comando 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, vedere [describe-replication-groups](#) nel AWS CLI Reference.

## Ridimensionamento di un gruppo di OSS replica Valkey o Redis () ElastiCache API

Il processo seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo di nodo più piccolo utilizzando il. ElastiCache API Durante questo processo, ElastiCache aggiorna le DNS voci in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Valkey 7.2 e versioni successive o Redis OSS 5.0.5 e versioni successive, puoi scalare i cluster abilitati al failover automatico mentre il cluster continua a rimanere online e a soddisfare le richieste in arrivo. Nella OSS versione 4.0.10 e precedenti di Redis, potresti

notare una breve interruzione delle operazioni di lettura e scrittura sulle versioni precedenti dal nodo principale mentre la voce viene aggiornata. DNS 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 ridimensionare un Valkey o Redis Replication Group () OSS ElastiCache API

1. Determina a quali tipi di nodi puoi ridimensionare utilizzando l' ElastiCache `ListAllowedNodeTypeModifications` azione 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 la pagina [ListAllowedNodeTypeModifications](#) nell'Amazon ElastiCache API Reference.

2. Scala il tuo attuale gruppo di replica fino al nuovo tipo di nodo utilizzando l'`ModifyRedplicationGroup` ElastiCache API azione 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'`ListAllowedNodeTypeModifications` azione nel passaggio precedente.
  - `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 la pagina [ModifyReplicationGroup](#) nell'Amazon ElastiCache API Reference.

3. Se lo hai utilizzato `ApplyImmediately=true`, monitora lo stato del gruppo di replica utilizzando l' `ElastiCache APIDescribeReplicationGroups` azione 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 la pagina [DescribeReplicationGroups](#) nell'Amazon ElastiCache API Reference.

## Aumento della capacità di lettura

Per aumentare la capacità di lettura, aggiungi repliche di lettura (fino a un massimo di cinque) al tuo gruppo di replica Valkey o RedisOSS.

È possibile scalare la capacità di lettura del OSS cluster Valkey o Redis utilizzando la ElastiCache console, il o il. AWS CLI ElastiCache API Per ulteriori informazioni, consulta [Aggiungere una replica di lettura per Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

## Diminuzione della capacità di lettura

Per ridurre la capacità di lettura, elimina una o più repliche di lettura dal OSS cluster Valkey o Redis con repliche (chiamate gruppo di replica in/). 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 Valkey o Redis OSS \(modalità cluster disabilitata\)](#).

## Scalabilità dei cluster in Valkey o Redis OSS (modalità cluster abilitata)

Man mano che la domanda dei cluster cambia, potresti decidere di migliorare le prestazioni o ridurre i costi modificando il numero di shard nel cluster Valkey o Redis OSS (modalità cluster abilitata). 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.

Puoi determinare se i tuoi nodi sono sotto pressione in termini di memoria monitorando le seguenti metriche: `FreeableMemorySwapUsageBytesUseForCache`

- CPU o collo di bottiglia della rete:

Se si riscontrano problemi di latenza/throughput del cluster, è possibile aumentare le dimensioni per risolvere tali problemi.

È possibile monitorare i livelli di latenza e velocità effettiva monitorando le seguenti metriche: `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `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'utilizzo del cluster per determinare se è possibile scalare in sicurezza o meno utilizzando le seguenti metriche: `FreeableMemory`, `SwapUsage`, `BytesUseForCache`, `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `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 degrado che si verifica dipende dal normale CPU utilizzo e dai dati.

Esistono due modi per scalare il cluster Valkey o Redis OSS (modalità cluster abilitata): scalabilità 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 state riducendo le dimensioni e la capacità di memoria del cluster, mediante scalabilità verticale o ridotta, assicuratevi che la nuova configurazione disponga di memoria sufficiente per i dati e il sovraccarico di Valkey o Redis. OSS

Per ulteriori informazioni, consulta [Select cache node size](#).

## Indice

- [Resharding offline per Valkey o Redis \(modalità cluster abilitata\) OSS](#)
- [Resharding online per Valkey o Redis OSS \(modalità cluster abilitata\)](#)
  - [Aggiunta delle partizioni con il resharding online](#)
  - [Rimozione delle partizioni con il resharding online](#)
    - [Rimozione delle partizioni \(Console\)](#)
    - [Rimozione delle partizioni \(AWS CLI\)](#)
    - [Rimozione dei frammenti \(\) ElastiCache API](#)
  - [Ribilanciamento delle partizioni online](#)
    - [Ribilanciamento delle partizioni online \(Console\)](#)
    - [Ribilanciamento delle partizioni online \(AWS CLI\)](#)
    - [Ribilanciamento delle partizioni online \(ElastiCache API\)](#)

- [Ridimensionamento verticale online tramite la modifica del tipo di nodo](#)
  - [Dimensionamento verso l'alto online](#)
    - [Scalabilità dei cluster di OSS cache Valkey o Redis \(console\)](#)
    - [Scalabilità dei cluster di cache Valkey o Redis \(\) OSS AWS CLI](#)
    - [Scalabilità dei cluster di OSS cache Valkey o Redis \(\) ElastiCache API](#)
  - [Dimensionamento verso il basso online](#)
    - [Ridimensionamento dei cluster di OSS cache Valkey o Redis \(console\)](#)
    - [Ridimensionamento dei cluster di cache Valkey o Redis \(\) OSS AWS CLI](#)
    - [Ridimensionamento dei cluster di OSS cache Valkey o Redis \(\) ElastiCache API](#)

## Resharding offline per Valkey o Redis (modalità cluster abilitata) OSS

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. Quando si esegue il reshard e il ribilanciamento offline, oltre a modificare il numero di shard nel gruppo di replica, è possibile effettuare le seguenti operazioni:

### Note

Il resharding offline non è supportato sui cluster Valkey o Redis con il tiering dei dati abilitato. OSS Per ulteriori informazioni, consulta [Suddivisione dei dati su più livelli in ElastiCache](#).

- 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 i tuoi shard, il cluster Valkey o Redis (modalità cluster abilitata) è offline OSS

1. Crea un backup manuale del tuo cluster Valkey o Redis esistente. OSS 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 [Ricerca degli endpoint di connessione in ElastiCache](#).

Resharding online per Valkey o Redis OSS (modalità cluster abilitata)

Utilizzando il resharding online e il ribilanciamento degli shard con ElastiCache Valkey 7.2 o versione successiva o Redis OSS versione 3.2.10 o successiva, puoi scalare il tuo cluster Valkey o Redis OSS (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:

- Scalabilità orizzontale: aumenta la capacità di lettura e scrittura aggiungendo shard (gruppi di nodi) al cluster Valkey o Redis (modalità cluster abilitata) (gruppo di replica). OSS

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.

- Scalabilità integrata: riduci la capacità di lettura e scrittura, e quindi i costi, rimuovendo gli shard dal cluster Valkey o Redis OSS (abilitato in modalità cluster).
- Ribilanciamento: sposta gli spazi chiave tra gli shard del cluster Valkey o Redis OSS (abilitata in modalità cluster) 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

- Questi processi richiedono Valkey 7.2 e versioni successive o la versione 3.2.10 o successiva del motore Redis. OSS Per ulteriori informazioni sull'aggiornamento della versione del motore, consultare [Gestione delle versioni per ElastiCache](#).
- 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 Con Valkey 7.2 e versioni successive e con Redis OSS versione 7 e successive: quando si ridimensiona il cluster, ElastiCache replicherà automaticamente le funzioni caricate in uno dei nodi esistenti (selezionati a caso) sui nuovi nodi. Se la tua applicazione utilizza [Functions](#), ti consigliamo di caricare tutte le funzioni su tutti gli shard prima di eseguire la scalabilità orizzontale, in modo che il cluster non finisca con definizioni di funzioni diverse su shard diversi.

Per ulteriori informazioni, consulta [Ridimensionamento di cluster online](#).

È possibile scalare o ribilanciare orizzontalmente i cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando, the e. AWS Management Console AWS CLI ElastiCache API



## Aggiunta delle partizioni con il resharding online

È possibile aggiungere shard al cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando, o, AWS Management Console AWS CLI ElastiCache API. Quando aggiungi shard a un cluster Valkey o Redis OSS (modalità cluster abilitata), tutti i tag sugli shard esistenti vengono copiati nei nuovi shard.

### Aggiunta delle partizioni (Console)

È possibile utilizzare il AWS Management Console per aggiungere uno o più shard al cluster Valkey o Redis OSS (modalità cluster abilitata). Questo processo viene descritto di seguito.

Per aggiungere shard al cluster Valkey o Redis OSS (modalità cluster abilitata)

1. Apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
3. Individua e scegli il nome, non la casella a sinistra del nome del cluster, del cluster Valkey o Redis OSS (modalità cluster abilitata) a cui desideri aggiungere gli shard.

#### Tip

Valkey o Redis OSS (modalità cluster abilitata) mostrano Clustered Valkey o Clustered Redis nella colonna Mode OSS

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)

Il processo seguente descrive come riconfigurare gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) aggiungendo shard utilizzando. 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` Facoltativo. 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 riconfigura gli spazi chiave su quattro shard in un cluster Valkey o Redis OSS (abilitato in modalità cluster) denominato `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 `-configuration` nella documentazione. `modify-replication-group-shard` AWS CLI](#)

## Aggiunta di partizioni (ElastiCache API)

È possibile utilizzare il ElastiCache API per riconfigurare online gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) 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` Facoltativo. 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 Valkey o Redis OSS (modalità cluster abilitata) aggiungendo shard utilizzando. ElastiCache API

### Example - Aggiunta dle partizioni

L'esempio seguente aggiunge gruppi di nodi al cluster Valkey o Redis OSS (modalità cluster abilitata) `my-cluster`, quindi ci sono un totale di quattro gruppi di nodi al termine dell'operazione. 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, vedere [ModifyReplicationGroupShardConfiguration](#) nel Reference. ElastiCache API

### Rimozione delle partizioni con il resharding online

È possibile rimuovere gli shard dal cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando AWS Management Console, AWS CLI o ElastiCache API

### Argomenti

- [Rimozione delle partizioni \(Console\)](#)
- [Rimozione delle partizioni \(AWS CLI\)](#)
- [Rimozione dei frammenti \(\) ElastiCache API](#)

### Rimozione delle partizioni (Console)

Il processo seguente descrive come riconfigurare gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) rimuovendo gli shard utilizzando AWS Management Console

Prima di rimuovere i gruppi di nodi (shard) dal gruppo di replica, assicurati che tutti i dati ElastiCache 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 Valkey o Redis OSS (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 gli shard dal cluster Valkey o Redis OSS (modalità cluster abilitata)

1. Apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
3. Individua e scegli il nome, non la casella a sinistra del nome del cluster, del cluster Valkey o Redis OSS (modalità cluster abilitata) da cui desideri rimuovere gli shard.


 Tip

I cluster Valkey o Redis OSS (modalità cluster abilitata) hanno un valore pari o superiore a 1 nella colonna Shards.

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)

Il processo seguente descrive come riconfigurare gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) rimuovendo gli shard utilizzando AWS CLI

 Important

Prima di rimuovere i gruppi di nodi (shard) dal gruppo di replica, assicurati che tutti i dati ElastiCache 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 Valkey o Redis OSS (modalità cluster abilitata). Non puoi rimuovere tutti 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 di 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 di partizioni.
- `--node-groups-to-remove` : Obbligatorio quando `--node-group-count` è inferiore al numero corrente di gruppi di nodi (partizioni). Un elenco di shard (gruppo di nodi) da rimuovere dal gruppo IDs di replica.

La procedura seguente descrive l'eliminazione di una o più partizioni.

#### Example - Rimozione delle partizioni

L'esempio seguente rimuove due gruppi di nodi dal cluster Valkey o Redis OSS (modalità cluster abilitata) `my-cluster`, quindi ci sono un totale di due gruppi di nodi al termine dell'operazione. 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 dei frammenti () ElastiCache API

È possibile utilizzare il ElastiCache API per riconfigurare online gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando l'operazione.

#### ModifyReplicationGroupShardConfiguration

Il processo seguente descrive come riconfigurare gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) rimuovendo gli shard utilizzando. ElastiCache API

**⚠ Important**

Prima di rimuovere i gruppi di nodi (shard) dal gruppo di replica, assicurati che tutti i dati ElastiCache 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.

È possibile utilizzare il ElastiCache API per rimuovere uno o più shard dal cluster Valkey o Redis OSS (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 shard (gruppo di nodi) da rimuovere dal gruppo IDs di replica.

La procedura seguente descrive l'eliminazione di una o più partizioni.

**Example - Rimozione delle partizioni**

L'esempio seguente rimuove due gruppi di nodi dal cluster Valkey o Redis OSS (modalità cluster abilitata) `my-cluster`, quindi ci sono un totale di due gruppi di nodi al termine dell'operazione. 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

È possibile ribilanciare gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando, o. AWS Management Console AWS CLI ElastiCache API

### Argomenti

- [Ribilanciamento delle partizioni online \(Console\)](#)
- [Ribilanciamento delle partizioni online \(AWS CLI\)](#)
- [Ribilanciamento delle partizioni online \(ElastiCache API\)](#)

## Ribilanciamento delle partizioni online (Console)

Il processo seguente descrive come riconfigurare gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) ribilanciando gli shard utilizzando. AWS Management Console

Per ribilanciare gli spazi chiave tra gli shard sul cluster Valkey o Redis (modalità cluster abilitata) OSS

1. Apri ElastiCache <https://console.aws.amazon.com/elasticache/> la console all'indirizzo.
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
3. Scegli il nome, non la casella a sinistra del nome, del cluster Valkey o Redis OSS (modalità cluster abilitata) che desideri ribilanciare.

 Tip

I cluster Valkey o Redis OSS (modalità cluster abilitata) hanno un valore pari o superiore a 1 nella colonna Shards.

4. Scegliere Rebalance (Ribilancia).
5. Quando richiesto, scegliere Rebalance (Ribilancia). Potresti vedere un messaggio simile a questo: *Slots in the replication group are uniformly distributed. Nothing to do. (Service: AmazonElastiCache; Status Code: 400; Error Code: InvalidReplicationGroupState; Request ID: 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)*. In tal caso, scegli 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.

Il processo seguente descrive come riconfigurare gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) ribilanciando gli shard utilizzando AWS CLI

### Example - Ribilanciamento delle partizioni in un cluster

L'esempio seguente ribilancia gli slot nel cluster Valkey o Redis OSS (modalità cluster abilitata) in modo che gli slot siano distribuiti nel modo più equo possibile. `my-cluster` 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 delle partizioni online (ElastiCache API)

È possibile utilizzare il ElastiCache API per riconfigurare online gli shard nel cluster Valkey o Redis OSS (modalità cluster abilitata) 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 tutti 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 Valkey o Redis OSS (modalità cluster abilitata) ribilanciando gli shard utilizzando il. ElastiCache API

#### Example - Ribilanciamento di un cluster

L'esempio seguente ribilancia gli slot nel cluster Valkey o Redis OSS (modalità cluster abilitata) in modo che gli slot siano distribuiti nel modo più equo possibile. `my-cluster` 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 la scalabilità verticale online con Valkey versione 7.2 o nuova o Redis OSS versione 3.2.10 o successiva, puoi scalare i tuoi cluster Valkey o Redis in modo dinamico con tempi di inattività minimi. OSS Ciò consente al cluster Valkey o Redis di soddisfare le richieste anche durante la scalabilità. OSS

### 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 [Suddivisione dei dati su più livelli in ElastiCache](#).

Puoi eseguire le operazioni indicate di seguito:

- Scalabilità verticale: aumenta la capacità di lettura e scrittura modificando il tipo di nodo del OSS cluster Valkey o Redis per utilizzare un tipo di nodo più grande.

ElastiCache ridimensiona dinamicamente il cluster rimanendo online e soddisfacendo le richieste.

- Riduzione verticale - Riduce la capacità di lettura e scrittura modificando il tipo di nodo affinché utilizzi un nodo più piccolo. Anche in questo caso, ridimensiona ElastiCache dinamicamente il cluster rimanendo online e soddisfacendo le 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 avere una capacità sufficiente ENI (Elastic Network Interface). 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 riservata per Valkey e Redis OSS](#).

- 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](#)
  - [Scalabilità dei cluster di OSS cache Valkey o Redis \(console\)](#)
  - [Scalabilità dei cluster di cache Valkey o Redis \(\) OSS AWS CLI](#)
  - [Scalabilità dei cluster di OSS cache Valkey o Redis \(\) ElastiCache API](#)
- [Dimensionamento verso il basso online](#)
  - [Ridimensionamento dei cluster di OSS cache Valkey o Redis \(console\)](#)
  - [Ridimensionamento dei cluster di cache Valkey o Redis \(\) OSS AWS CLI](#)
  - [Ridimensionamento dei cluster di OSS cache Valkey o Redis \(\) ElastiCache API](#)

## Dimensionamento verso l'alto online

### Argomenti

- [Scalabilità dei cluster di OSS cache Valkey o Redis \(console\)](#)
- [Scalabilità dei cluster di cache Valkey o Redis \(\) OSS AWS CLI](#)
- [Scalabilità dei cluster di OSS cache Valkey o Redis \(\) ElastiCache API](#)

## Scalabilità dei cluster di OSS cache Valkey o Redis (console)

La procedura seguente descrive come scalare un OSS cluster Valkey o Redis utilizzando la console di ElastiCache gestione. Durante questo processo, il cluster continuerà a soddisfare le richieste con tempi di inattività minimi.

### Per scalare un OSS cluster Valkey o Redis (console)

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
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.

## Scalabilità dei cluster di cache Valkey o Redis () OSS AWS CLI

La procedura seguente descrive come scalare un cluster di cache Valkey o Redis OSS utilizzando AWS CLI Durante questo processo, il cluster continuerà a soddisfare le richieste con tempi di inattività minimi.

### Per scalare un cluster di OSS cache Valkey o Redis ()AWS CLI

1. Determina i tipi di nodi fino a cui puoi scalare eseguendo il AWS CLI `list-allowed-node-type-modifications` comando 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 precedente ha un aspetto simile JSON al seguente (formato).

```
{
 "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 la pagina [list-allowed-node-type-modifiche](#) nel riferimento AWS CLI

2. Modificate il gruppo di replica per adattarlo al nuovo tipo di nodo più grande, utilizzando il AWS CLI `modify-replication-group` comando e 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 precedente ha un aspetto simile al seguente (JSONformato).

```
{
 "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 la pagina [modify-replication-group](#) nel AWS CLI Reference.

3. Se hai usato il `--apply-immediately`, controlla lo stato del cluster di cache usando il AWS CLI `describe-cache-clusters` comando con il seguente parametro. Quando lo stato cambia in disponibile, è possibile iniziare a utilizzare il nuovo cluster di cache più grande.

## Scalabilità dei cluster di OSS cache Valkey o Redis () ElastiCache API

Il seguente processo ridimensiona il cluster di cache dal tipo di nodo corrente a un nuovo tipo di nodo più grande utilizzando il. ElastiCache API Durante questo processo, ElastiCache aggiorna le DNS voci in modo che puntino ai nuovi nodi. Per questo motivo non è necessario aggiornare gli endpoint nell'applicazione. Per Valkey 7.2 e versioni successive Redis OSS 5.0.5 e versioni successive, puoi scalare i cluster abilitati al failover automatico mentre il cluster continua a rimanere online e a soddisfare le richieste in arrivo. Nella versione Redis OSS 4.0.10 e precedenti, è possibile notare una breve interruzione delle operazioni di lettura e scrittura sulle versioni precedenti dal nodo principale durante l'aggiornamento della voce. DNS

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 scalare un Valkey o Redis Cache Cluster () OSS ElastiCache API

1. Determina a quali tipi di nodi puoi scalare utilizzando l' ElastiCache `APIListAllowedNodeTypeModifications` azione 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 la pagina [ListAllowedNodeTypeModifications](#) nell'Amazon ElastiCache API Reference.

2. Scala il tuo attuale gruppo di replica fino al nuovo tipo di nodo utilizzando l'`ModifyReplicationGroup` ElastiCache API azione 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'`ListAllowedNodeTypeModifications` azione nel passaggio precedente.
- **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 la pagina [ModifyReplicationGroup](#) nell'Amazon ElastiCache API Reference.

3. Se lo hai utilizzato `ApplyImmediately=true`, monitora lo stato del gruppo di replica utilizzando l' `ElastiCache APIDescribeReplicationGroups` azione 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 la pagina [DescribeReplicationGroups](#) nell'Amazon ElastiCache API Reference.

## Dimensionamento verso il basso online

### Argomenti

- [Ridimensionamento dei cluster di OSS cache Valkey o Redis \(console\)](#)
- [Ridimensionamento dei cluster di cache Valkey o Redis \(\) OSS AWS CLI](#)
- [Ridimensionamento dei cluster di OSS cache Valkey o Redis \(\) ElastiCache API](#)

### Ridimensionamento dei cluster di OSS cache Valkey o Redis (console)

La procedura seguente descrive come ridimensionare un OSS cluster Valkey o Redis utilizzando la console di ElastiCache gestione. Durante questo processo, il OSS cluster Valkey o Redis continuerà a soddisfare le richieste con tempi di inattività minimi.

### Per ridimensionare un cluster Valkey o Redis OSS (console)

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
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.

## Ridimensionamento dei cluster di cache Valkey o Redis () OSS AWS CLI

La procedura seguente descrive come ridimensionare un cluster di cache Valkey o Redis OSS utilizzando AWS CLI. Durante questo processo, il cluster continuerà a soddisfare le richieste con tempi di inattività minimi.

### Per ridimensionare un cluster di OSS cache Valkey o Redis ()AWS CLI

1. Determina i tipi di nodi a cui puoi ridimensionare eseguendo il AWS CLI `list-allowed-node-type-modifications` comando 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 precedente ha un aspetto simile JSON al seguente (formato).

```
{
 "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 la pagina [list-allowed-node-type-modifiche](#) nel riferimento.AWS CLI

2. Modificate il gruppo di replica per ridurlo al nuovo tipo di nodo più piccolo, utilizzando il AWS CLI `modify-replication-group` comando e 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 precedente ha un aspetto simile al seguente (JSONformato).

```
{
 "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 la pagina [modify-replication-group](#) nel AWS CLI Reference.

3. Se hai usato il `--apply-immediately`, controlla lo stato del cluster di cache usando il AWS CLI `describe-cache-clusters` comando 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 OSS cache Valkey o Redis () ElastiCache API

Il processo seguente ridimensiona il gruppo di replica dal tipo di nodo corrente a un nuovo tipo di nodo più piccolo utilizzando il ElastiCache API. Durante questo processo, il OSS cluster Valkey o Redis continuerà a soddisfare le richieste con tempi di inattività minimi.

Il tempo necessario per diminuire utilizzando un tipo di nodo di minori dimensioni varia a seconda del tipo di nodo e della quantità di dati nel cluster di cache attuale.



## Scalabilità verso il basso () ElastiCache API

1. Determina a quali tipi di nodi puoi ridimensionare utilizzando l' ElastiCache API `ListAllowedNodeTypeModifications` azione 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 la pagina [ListAllowedNodeTypeModifications](#) nell'Amazon ElastiCache API Reference.

2. Ridimensiona il tuo attuale gruppo di replica fino al nuovo tipo di nodo utilizzando l'`ModifyReplicationGroup` ElastiCache API azione 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'`ListAllowedNodeTypeModifications` azione nel passaggio precedente.
- `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 la pagina [ModifyReplicationGroup](#) nell'Amazon ElastiCache API Reference.

## Guida introduttiva JSON a Valkey e Redis OSS

ElastiCache supporta il formato nativo JavaScript Object Notation (JSON), un modo semplice e senza schemi per codificare set di dati complessi all'interno di cluster Valkey e Redis. OSS È possibile archiviare e accedere in modo nativo ai dati utilizzando il formato JavaScript Object Notation (JSON) all'interno dei cluster e aggiornare i JSON dati archiviati in tali cluster, senza dover gestire codice personalizzato per serializzarli e deserializzarli.

Oltre a utilizzare le OSS API operazioni Valkey e Redis per le applicazioni che funzionano su piùJSON, ora è possibile recuperare e aggiornare in modo efficiente porzioni specifiche di un documento senza dover manipolare l'intero oggetto. JSON Ciò può migliorare le prestazioni e ridurre i costi. [È inoltre possibile cercare il contenuto del JSON documento utilizzando l'interrogazione in stile Goessner.](#) JSONPath

Dopo aver creato un cluster con una versione del motore supportata, il tipo di JSON dati e i comandi associati sono automaticamente disponibili. APIcompatibile e RDB compatibile con la versione 2 del JSON modulo, in modo da poter migrare facilmente le applicazioni Valkey e Redis JSON OSS basate su esistenti. ElastiCache Per ulteriori informazioni sui comandi supportati, vedere. [Comandi Valkey e Redis OSS supportati](#)

Le metriche JSON `JsonBasedCmds` relative `JsonBasedCmdsLatency` sono incorporate CloudWatch per monitorare l'utilizzo di questo tipo di dati. Per ulteriori informazioni, consulta [Metriche per Valkey](#) e Redis. OSS

### Note

Per utilizzarlo JSON, è necessario utilizzare Valkey 7.2 e versioni successive oppure la versione 6.2.6 o successiva del motore Redis OSS.

## Argomenti

- [JSON panoramica dei tipi di dati](#)
- [Comandi Valkey e Redis OSS supportati](#)

## JSON panoramica dei tipi di dati

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

## Terminology

Termine	Descrizione
JSONdocumento	Si riferisce al valore di una JSON chiave.
JSONvalore	Si riferisce a un sottoinsieme di un JSON documento, inclusa la radice che rappresenta l'intero documento. Un valore può essere un container o una voce vuota in un container.
JSONelemento	Equivalente al JSON valore.

## JSONStandard supportato

JSONil formato è conforme agli standard di interscambio [RFCdati 7159](#) e [ECMA-404](#)JSON. UTF-8 È supportato il formato [Unicode](#) nel testo. JSON

## Elemento radice

L'elemento radice può essere di qualsiasi tipo di JSON dati. Si noti che nella versione precedente del RFC 4627, solo gli oggetti o gli array erano consentiti come valori radice. Dall'aggiornamento alla versione RFC 7159, la radice di un JSON documento può essere di qualsiasi tipo JSON di dati.

## Limite delle dimensioni del documento

JSONi documenti vengono archiviati internamente in un formato ottimizzato per un accesso e una modifica rapidi. Questo formato generalmente comporta un consumo di memoria lievemente superiore alla rappresentazione serializzata equivalente dello stesso documento.

Il consumo di memoria da parte di un singolo JSON documento è limitato a 64 MB, che è la dimensione della struttura dei dati in memoria, non della JSON stringa. È possibile controllare la quantità di memoria consumata da un JSON documento utilizzando il `JSON.DEBUG MEMORY` comando.

## JSON ACLs

- Analogamente alle categorie esistenti per tipo di dati (`@string`, `@hash`, ecc.), viene aggiunta una nuova categoria `@json` per semplificare la gestione dell'accesso a comandi e dati. JSON Nessun altro OSS comando Valkey o Redis esistente appartiene alla categoria `@json`. Tutti i JSON comandi applicano eventuali restrizioni e autorizzazioni relative allo spazio dei tasti o ai comandi.
- Esistono cinque OSS ACL categorie Valkey e Redis esistenti che vengono aggiornate per includere i nuovi JSON comandi: `@read`, `@write`, `@fast`, `@slow` e `@admin`. La tabella seguente indica la mappatura dei JSON comandi alle categorie appropriate.

## ACL

JSONcomando	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		

JSONcomando	@read	@write	@fast	@slow	@admin
JSON.ARRINSERT		y	y		
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		
JSON.NUMMULTBY		y	y		
JSON.OBJECTS	y		y		

JSONcomando	@read	@write	@fast	@slow	@admin
JSON.OBJLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRAPPEND		y	y		
JSON.STRLLEN	y		y		
JSON.STRLLEN	y		y		
JSON.TOGGLE		y	y		
JSON.TYPE	y		y		
JSON.NUMINCRBY		y	y		

## Limite di profondità di nidificazione

Quando un JSON oggetto o un array ha un elemento che è a sua volta un altro JSON oggetto o array, si dice che quell'oggetto o array interno «nidifica» all'interno dell'oggetto o dell'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 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

Redis JSON supporta due tipi di sintassi dei percorsi:

- Sintassi avanzata: segue la JSONPath sintassi descritta da [Goessner](#), come illustrato 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
\$	Lelemento 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.

Simbolo/espressione	Descrizione
[,]	Operatore di unione.
[start:end:step]	Operatore di sezionamento di un array.
?()	Applica un'espressione filtro (script) all'array o all'oggetto corrente.
()	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.
&&	LogicoAND, utilizzato per combinare più espressioni di filtro.
	OR logico, utilizzato per combinare più espressioni filtro.

## Examples (Esempi)

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

```
{ "store": {
```



```
"book": [
 { "category": "reference",
 "author": "Nigel Rees",
 "title": "Sayings of the Century",
 "price": 8.95,
 "in-stock": true,
 "sold": true
 },
 { "category": "fiction",
 "author": "Evelyn Waugh",
 "title": "Sword of Honour",
 "price": 12.99,
 "in-stock": false,
 "sold": true
 },
 { "category": "fiction",
 "author": "Herman Melville",
 "title": "Moby Dick",
 "isbn": "0-553-21311-3",
 "price": 8.99,
 "in-stock": true,
 "sold": false
 },
 { "category": "fiction",
 "author": "J. R. R. Tolkien",
 "title": "The Lord of the Rings",
 "isbn": "0-395-19395-8",
 "price": 22.99,
 "in-stock": false,
 "sold": false
 }
],
"bicycle": {
 "color": "red",
 "price": 19.95,
 "in-stock": true,
 "sold": false
}
}
```

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.
<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 ISBN numero.
<code>\$.book[?(@.price&lt;10)]</code>	Tutti i libri che costano meno di 10 euro.
<code>'\$.book[?(@.price &lt; 10)]'</code>	Tutti i libri che costano meno di 10 euro. (Il percorso deve essere tra virgolette se contiene spazi vuoti.)
<code>'\$.book[?(@["price"] &lt; 10)]'</code>	Tutti i libri che costano meno di 10 euro.
<code>'\$.book[?(@.["price"] &lt; 10)]'</code>	Tutti i libri che costano meno di 10 euro.

Path	Descrizione
<code>\$.book[?(@.price&gt;=10&amp;&amp;@.price&lt;=100)]</code>	Tutti i libri nella fascia di prezzo da 10 a 100 euro, inclusi.
<code>'\$.book[?(@.price&gt;=10 &amp;&amp; @.price&lt;=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.
<code>'\$.book[?(@.sold == true    @.in-stock == false)]'</code>	Tutti i libri venduti o esauriti. (Il percorso deve essere tra virgolette se contiene spazi vuoti.)
<code>'\$.store.book[?(@.["category"] == "fiction")]</code>	Tutti i libri della categoria fiction.
<code>'\$.store.book[?(@.["category"] != "fiction")]</code>	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 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.

## JSON-metriche correlate

Vengono fornite le seguenti metriche JSON informative:

Info	Descrizione
json_total_memory_bytes	Memoria totale allocata agli oggetti. JSON
json_num_documents	Numero totale di documenti in Valkey o Redis. OSS

Per interrogare le metriche principali, esegui il seguente comando:

```
info json_core_metrics
```

## Come ElastiCache interagisce con Valkey e Redis OSS JSON

La sezione seguente descrive come ElastiCache con Valkey e Redis OSS interagisce con il tipo di dati. JSON

### Precedenza degli operatori

Durante la valutazione delle espressioni condizionali per il filtro, `&&` hanno la precedenza, quindi vengono valutati `||`, come nella maggior parte dei linguaggi. Operazioni tra parentesi vengono eseguite per prime.

### Comportamento del limite massimo di nidificazione dei percorsi

Il limite massimo di annidamento dei percorsi in ElastiCache (RedisOSS) è 128. Per cui, un valore come `$ . a . b . c . d . . .` può raggiungere solo 128 livelli.

### Gestione dei valori numerici

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

### Rappresentazioni numeriche:

Quando un JSON numero viene ricevuto in input, viene convertito in una delle due rappresentazioni binarie interne: un intero con segno a 64 bit o una virgola mobile a IEEE doppia precisione a 64 bit. La stringa originaria e tutta la formattazione non vengono mantenute. Pertanto, quando un numero viene emesso come parte di una JSON risposta, 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 numeri interi e il risultato non rientra nell'intervallo di `int64`, diventa automaticamente un numero in virgola mobile a IEEE doppia precisione a 64 bit.
- Se almeno uno dei numeri è a virgola mobile, il risultato è un numero in virgola mobile a IEEE doppia precisione a 64 bit.
- Se il risultato supera l'intervallo del IEEE doppio a 64 bit, il comando restituisce un errore. **OVERFLOW**

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

## Filtraggio array diretto

ElastiCache con Valkey o Redis OSS filtra direttamente gli oggetti dell'array.

Per dati come `[0, 1, 2, 3, 4, 5, 6]` e una query di percorso come `[$[?(@<4)]]`, o dati simili `{"my_key": [0, 1, 2, 3, 4, 5, 6]}` e una query di percorso come `$.my_key[?(@<4)]`, ElastiCache con Valkey o Redis OSS restituirebbe `[1,2,3]` in entrambe le circostanze.

## Comportamento di indicizzazione array

ElastiCache con Valkey o Redis OSS consente indici positivi e 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 con Valkey o Redis OSS non arrotonda gli indici degli array verso il basso o verso l'alto, quindi se si dispone di un array con una lunghezza di 5, chiamare l'indice 5 o superiore, oppure -6 o inferiore, non produrrebbe alcun risultato.

## Valutazione della sintassi rigida

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

## Comandi Valkey e Redis OSS supportati

ElastiCache supporta i seguenti comandi Valkey e Redis OSSJSON:

### 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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (obbligatorio) — Un JSON percorso.
- **json** (richiesto) — Il JSON valore da aggiungere 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.
- SYNTAXERrore se uno degli argomenti json di input non è una stringa valida. JSON
- 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.
- SYNTAXERrore se uno degli argomenti json di input non è una stringa validaJSON.
- Errore NONEXISTENT se il percorso non esiste.

Examples (Esempi)

Sintassi avanzata del percorso:

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

Sintassi limitata del percorso:

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

## JSON.ARRINDEX

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

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

## Sintassi

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

- **key** (obbligatorio) — Una chiave Valkey o Redis OSS di tipo documento. JSON
- **path** (obbligatorio) — Un JSON percorso.
- **json-scalar** (richiesto) — Il valore scalare per cercare. JSONscalare si riferisce a valori che non sono oggetti o matrici. 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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (obbligatorio) — Un JSON percorso.
- **index** (richiesto) — Un indice array prima del quale sono inseriti valori.
- **json** (richiesto) — Il JSON valore da aggiungere 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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (opzionale) — Un JSON percorso. 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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (opzionale) — Un JSON percorso. 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 visualizzatoJSON.
- 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` (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- `path` (obbligatorio) — Un JSON percorso.
- `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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (opzionale) — Un JSON percorso. 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>[percorso]** — Riporta l'utilizzo della memoria in byte di un JSON valore. Se non è fornito, viene ripristinato il percorso predefinito, la radice.
- **FIELDS<key>[percorso]** — Riporta il numero di campi nel percorso del documento specificato. Se non è fornito, viene ripristinato il percorso predefinito, la radice. Ogni JSON valore non contenitore conta come un campo. Gli oggetti e gli array contano ricorsivamente un campo per ciascuno dei valori che li contengono. JSON Ogni valore container, tranne il container radice, viene conteggiato come un campo aggiuntivo.
- **HELP**— Stampa i messaggi di aiuto del comando.

### Sintassi

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

Dipende dal sottocomando:

### MEMORY

- Se il percorso è una sintassi avanzata:
  - Restituisce una matrice di numeri interi che rappresentano la dimensione della memoria (in byte) del JSON valore in ogni percorso.
  - Restituisce un array vuoto se la chiave Valkey o Redis non esiste OSS.

- Se il percorso è una sintassi limitata:
  - Restituisce un numero intero, la dimensione della memoria e il JSON valore in byte.
  - Restituisce null se la chiave Valkey o Redis non esisteOSS.

## FIELDS

- Se il percorso è una sintassi avanzata:
  - Restituisce una matrice di numeri interi che rappresentano il numero di campi di JSON valore in ogni percorso.
  - Restituisce un array vuoto se la chiave Valkey o Redis non OSS esiste.
- Se il percorso è una sintassi limitata:
  - Restituisce un numero intero, il numero di campi del valore. JSON
  - Restituisce null se la chiave Valkey o Redis non esisteOSS.

HELP— Restituisce una serie di messaggi di aiuto.

## Examples (Esempi)

Sintassi avanzata del percorso:

```
127.0.0.1:6379> JSON.SET k1 . '[1, 2.3, "foo", true, null, {}, [], {"a":1, "b":2},
[1,2,3]]'
OK
127.0.0.1:6379> JSON.DEBUG MEMORY k1 $[*]
1) (integer) 16
2) (integer) 16
3) (integer) 19
4) (integer) 16
5) (integer) 16
6) (integer) 16
7) (integer) 16
8) (integer) 50
9) (integer) 64
127.0.0.1:6379> JSON.DEBUG FIELDS k1 $[*]
1) (integer) 1
2) (integer) 1
3) (integer) 1
4) (integer) 1
5) (integer) 1
```

```
6) (integer) 0
7) (integer) 0
8) (integer) 2
9) (integer) 3
```

### Sintassi limitata del percorso:

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

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

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

## JSON.DEL

Elimina i JSON valori nel percorso in una chiave del documento. Se il percorso è la radice, equivale a eliminare la chiave da Valkey o Redis. OSS

### Sintassi

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

- **key** (obbligatorio) — Una chiave Valkey o Redis OSS di tipo documento. JSON
- **path** (opzionale) — Un JSON percorso. Se non è fornito, per impostazione predefinita viene ripristinata la radice.

### Valori restituiti

- Numero di elementi eliminati.
- 0 se la chiave Valkey o Redis OSS non esiste.
- 0 se il JSON percorso 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 serializzato JSON in uno o più percorsi.

### Sintassi

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

- **key** (obbligatorio) — Una chiave Valkey o Redis di JSON tipo OSS documento.
- **INDENT/NEWLINE/SPACE**(opzionale) — Controlla il formato della JSON stringa restituita, ovvero «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, può essere presente per motivi di compatibilità con le versioni precedenti e non ha altri effetti.
- **path** (opzionale) — Zero o più JSON percorsi, il valore predefinito è root se non ne viene fornito nessuno. Gli argomenti del percorso devono essere collocati alla fine.

### Valori restituiti

Sintassi avanzata del percorso:

Se viene fornito un percorso:

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

Se vengono forniti più percorsi:

- Restituisce un JSON oggetto con stringhe, 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.*
["\t21 2nd Street\t","\tNew York\t","\tNY\t","\t10021-3100\t"]
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
["\n\t\t21 2nd Street\t","\n\t\tNew York\t","\n\t\tNY\t","\n\t\t10021-3100\t\n"]
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
{"\t$.firstName\t":["\tJohn\t"],"\t$.lastName\t":["\tSmith\t"],"\t$.age\t":["\t27\t"]}
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 $.*
["{ },{\t"a\t":1},{\t"a\t":1,\t"b\t":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
{"\tstreet\t":["\t21 2nd Street\t"],"\tcity\t":["\tNew York\t"],"\tstate\t":["\tNY\t"],"\tzipcode\t":
\t10021-3100\t"}
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" .address
{"\n\t\tstreet\t": \t21 2nd Street\t","\n\t\tcity\t": \tNew York\t","\n\t\tstate\t": \tNY\t","\n
\t\tzipcode\t": \t10021-3100\t\n"}
```

```
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
"{\".firstName\": \"John\", \".lastName\": \"Smith\", \".age\": 27}"
```

## JSON.MGET

Viene serializzato JSONs nel percorso da più chiavi del documento. Restituisce null per una chiave o un percorso inesistente. JSON

### Sintassi

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

- chiave (obbligatoria): una o più chiavi Valkey o Redis OSS di tipo documento.
- path (obbligatorio) — Un JSON percorso.

### Valori restituiti

- Array di stringhe in blocco. La dimensione dell'array è uguale al numero di chiavi nel comando. Ogni elemento dell'array viene compilato con (a) il codice serializzato JSON in base al percorso o (b) null se la chiave non esiste, il percorso non esiste nel documento o il percorso non è valido (errore di sintassi).
- Se una delle chiavi specificate esiste e non è una JSON chiave, il comando restituisce un 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 (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- path (obbligatorio) — Un JSON percorso.
- 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.
- OVERFLOWerrore se il risultato non rientra nell'intervallo del IEEE doppio 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.
- OVERFLOWerrore se il risultato non rientra nell'intervallo del IEEE doppio a 64 bit.
- NONEXISTENT se la chiave del documento non esiste.

## Examples (Esempi)

Sintassi avanzata del percorso:

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

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

127.0.0.1:6379> JSON.SET k2 $ '{"a":{ }, "b":{"a":1}, "c":{"a":1, "b":2}, "d":{"a":1, "b":2, "c":3}}'
```

```

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

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

```

### Sintassi limitata del percorso:

```

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

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

```

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

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

127.0.0.1:6379> JSON.SET k3 . '{"a":{"a":"a"}, "b":{"a":"a", "b":1}, "c":{"a":"a",
 "b":"b"}, "d":{"a":1, "b":"b", "c":3}}'
OK
127.0.0.1:6379> JSON.NUMINCRBY k3 .a.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .b.* 1
"2"
127.0.0.1:6379> JSON.NUMINCRBY k3 .c.* 1
(error) WRONGTYPE JSON element is not a number
127.0.0.1:6379> JSON.NUMINCRBY k3 .d.* 1
"4"
```

## JSON.NUMMULTBY

Moltiplica i valori numerici nel percorso di un determinato numero.

### Sintassi

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

- **key** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (obbligatorio) — Un JSON percorso.
- **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.
- `OVERFLOW` errore se il risultato non rientra nell'intervallo di un numero in virgola mobile a IEEE doppia precisione 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.
- `OVERFLOW` errore se il risultato non rientra nell'intervallo di un valore IEEE doppio 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\":{\"a\":2},\"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\":{\"a\":2},\"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\":{\"a\":2},\"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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (opzionale) — Un JSON percorso. 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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (opzionale) — Un JSON percorso. 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 JSON valore nel percorso specificato in Valkey o Redis OSS Serialization Protocol ().  
RESP Se il valore è container, la risposta è una RESP matrice o una matrice annidata.

- JSONnull è mappato alla RESP Null Bulk String.
- JSONI valori booleani vengono mappati alle rispettive stringhe semplici. RESP
- I numeri interi vengono mappati su numeri interi. RESP
- I numeri a virgola mobile IEEE doppia a 64 bit vengono mappati su Bulk Strings. RESP
- JSONle stringhe sono mappate su Bulk Strings. RESP
- JSONgli array sono rappresentati come RESP Array, dove il primo elemento è la semplice stringa [, seguita dagli elementi dell'array.
- JSONgli oggetti sono rappresentati come RESP Array, dove il primo elemento è la semplice stringa {, seguita da coppie chiave-valore, ognuna delle quali è una stringa di massa. RESP

## Sintassi

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

- key (obbligatorio) — Una chiave Valkey o Redis OSS di tipo documento. JSON
- path (opzionale) — Un JSON percorso. 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 RESP forma del valore in un percorso.
- Array vuoto se la chiave del documento non esiste.

Se il percorso è una sintassi limitata:

- Array che rappresenta la RESP forma 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 JSON i valori sul percorso.

Se il percorso richiede un membro oggetto:

- Se l'elemento principale non esiste, il comando restituisce un NONEXISTENT errore.

- 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 NONEXISTENT errore.
  - Se il membro esiste, il relativo valore verrà sostituito dal JSON valore.

Se il percorso richiede un indice di array:

- Se l'elemento principale non esiste, il comando restituisce un NONEXISTENT errore.
- Se l'elemento principale esiste ma non è un array, il comando restituisce ERROR.
- Se l'elemento principale esiste ma l'indice non è compreso nei limiti, il comando restituisce un OUTFBOUNDARIES errore.
- Se l'elemento principale esiste e l'indice è valido, l'elemento verrà sostituito dal nuovo JSON valore.

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

## Sintassi

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

[NX | XX] Dove è possibile avere 0 o 1 identificatori [NX | XX].

- **key** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (obbligatorio) — Un JSON percorso. Per una nuova chiave, il JSON percorso deve essere la radice «.».
- **NX** (opzionale): se il percorso è la radice, imposta il valore solo se la chiave 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** (opzionale): se il percorso è la radice, imposta il valore solo se la chiave 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) OUTFBOUNDARIES Array index is out of bounds
```

## JSON.STRAPPEND

Aggiunge una stringa alle JSON stringhe del percorso.

### Sintassi

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

- **key** (obbligatorio) — Una chiave Valkey o Redis OSS di tipo documento. JSON
- **path** (opzionale) — Un JSON percorso. Se non è fornito, per impostazione predefinita viene ripristinata la radice.
- **json\_string** (obbligatorio) — La JSON rappresentazione di una stringa. Nota che una JSON stringa deve essere citata. 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.
- **SYNTAXERR** errore se l'argomento json di input non è una stringa validaJSON.
- 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.
- **WRONGTYPE** errore se l'argomento json di input non è una stringa validaJSON.
- Errore **NONEXISTENT** se il percorso non esiste.

### Examples (Esempi)

Sintassi avanzata del percorso:

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

OK

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.a 'a'
```

1) (integer) 2

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.a.* 'a'
```

1) (integer) 3

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.b.* 'a'
```

1) (integer) 2  
2) (nil)

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.* 'a'
```

1) (integer) 2  
2) (integer) 3

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.c.b 'a'
```

1) (integer) 4

```
127.0.0.1:6379> JSON.STRAPPEND k1 $.d.* 'a'
```

1) (nil)  
2) (integer) 2  
3) (nil)

### Sintassi limitata del percorso:

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

OK

```
127.0.0.1:6379> JSON.STRAPPEND k1 .a.a 'a'
```

(integer) 2

```
127.0.0.1:6379> JSON.STRAPPEND k1 .a.* 'a'
```

(integer) 3

```
127.0.0.1:6379> JSON.STRAPPEND k1 .b.* 'a'
```

(integer) 2

```
127.0.0.1:6379> JSON.STRAPPEND k1 .c.* 'a'
```

(integer) 3

```
127.0.0.1:6379> JSON.STRAPPEND k1 .c.b 'a'
```

(integer) 4

```
127.0.0.1:6379> JSON.STRAPPEND k1 .d.* 'a'
```

(integer) 2

## JSON.STRLEN

Ottiene le lunghezze dei valori delle JSON stringhe nel percorso.

### Sintassi

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

- **key** (obbligatorio) — Una chiave Valkey o Redis di JSON tipo OSS documento.
- **path** (opzionale) — Un JSON percorso. 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** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (opzionale) — Un JSON percorso. 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 lotipo di valori nel percorso indicato.

### Sintassi

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

- **key** (obbligatorio) — Una chiave Valkey o Redis di OSS tipo JSON documento.
- **path** (opzionale) — Un JSON percorso. 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 JSON percorso 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
```

## Taggare le tue risorse ElastiCache

Per aiutarti a gestire i tuoi cluster e altre ElastiCache risorse, puoi assegnare i tuoi metadati a ciascuna risorsa sotto forma di tag. I tag consentono di classificare le AWS risorse in diversi modi, ad esempio per scopo, proprietario o ambiente. Questa caratteristica è molto utile quando hai tante risorse dello stesso tipo in quanto puoi rapidamente individuare una risorsa specifica in base ai tag assegnati. 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 si assegna a una AWS risorsa. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. I tag consentono di classificare le AWS risorse in diversi modi, ad esempio per scopo o proprietario. Ad esempio, puoi definire un set di tag per i ElastiCache cluster del tuo account che ti aiutino a tenere traccia del proprietario e del gruppo di utenti di ogni istanza.

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 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.

È possibile lavorare con i tag utilizzando i AWS Management Console AWS CLI, i ElastiCache API e.

Se lo utilizzi IAM, puoi controllare quali utenti del tuo AWS account sono autorizzati a creare, modificare o eliminare i tag. Per ulteriori informazioni, consulta [Autorizzazioni a livello di risorsa](#).

## Risorse cui è possibile associare tag

Puoi taggare la maggior parte ElastiCache delle risorse già esistenti nel tuo account. Nella tabella seguente sono elencate le risorse che supportano il tagging. Se utilizzi il AWS Management Console, puoi applicare tag alle risorse utilizzando il [Tag Editor](#). 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 tag Name, ma questo tag non ha alcun significato semantico per il ElastiCache servizio.

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 Amazon ElastiCache API, the o an AWS CLI AWS SDK, puoi utilizzare il Tags parametro nell' ElastiCache API azione pertinente per applicare i tag. Questi sono:

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`


- `CreateUserGroup`
- `CreateUser`
- `PurchaseReservedCacheNodesOffering`

La tabella seguente descrive le ElastiCache risorse che possono essere taggate e le risorse che possono essere taggate al momento della creazione utilizzando il ElastiCache API AWS CLI, il o un AWS SDK.

Supporto per ElastiCache l'etichettatura delle risorse

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 a livello di risorsa basate su tag nelle tue IAM politiche alle ElastiCache API azioni che supportano l'aggiunta di tag durante la creazione per implementare un controllo granulare sugli utenti e i gruppi che possono taggare le risorse al momento della 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.
- AddTagsToResourcee: RemoveTagsFromResource
  - I tag verranno aggiunti/rimossi dal gruppo di replica e l'operazione verrà propagata a tutti i cluster del gruppo di replica.

#### Note

AddTagsToResourcee RemoveTagsFromResource non può essere utilizzato per parametri e gruppi di sicurezza predefiniti.

- IncreaseReplicaCounte 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.
- Lunghezza massima della chiave: 128 caratteri Unicode in UTF -8.
- Lunghezza massima del valore: 256 caratteri Unicode in UTF -8.
- Sebbene ElastiCache consenta qualsiasi carattere nei tag, altri servizi possono essere restrittivi. I caratteri consentiti in tutti i servizi sono: lettere, numeri e spazi rappresentabili in UTF -8 e i seguenti caratteri: + - =. \_:/@
- Per le chiavi e i valori dei tag viene fatta la distinzione tra maiuscole e minuscole.

- Il `aws` : prefisso è riservato all'uso 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 ElastiCache risorse che puoi taggare, consulta [Risorse cui è possibile associare tag](#).

## Esempio: assegnazione di tag alle risorse

- Creazione di una cache serverless utilizzando i tag. Questo esempio utilizza Memcached come motore.

```
aws elasticache create-serverless-cache \
 --serverless-cache-name CacheName \
 --engine memcached \
 --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 \
 --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

```
--engine valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Creazione di un cluster di cache con i tag. Questo esempio utilizza Redis come motore.

```
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 valkey \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Creazione di uno snapshot serverless con tag. Questo esempio utilizza Memcached come motore.

```
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

Le istantanee sono attualmente disponibili solo per Redis. 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 AddTagsToResource l'azione su 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 CopySnapshot l'azione se lo snapshot di origine ha il tag Project= XYZ e il tag di richiesta 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 tag di allocazione dei costi alle tue risorse in Amazon ElastiCache, puoi tenere traccia dei costi raggruppando le spese sulle fatture in base ai valori dei tag di risorsa.

Un tag di allocazione ElastiCache dei costi è una coppia chiave-valore che definisci e associ 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.

Utilizzate i tag di allocazione dei costi per organizzare la AWS fattura in modo da rispecchiare la vostra struttura dei costi. A tale scopo, registrati per ricevere una fattura sul tuo AWS account con i valori chiave dell'etichetta inclusi. 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)`. Potrai quindi visualizzare i `ElastiCache` costi totali suddivisi per regione. Per ulteriori informazioni, consulta la pagina sull'[utilizzo dei tag per l'allocazione dei costi](#) nella Guida per l'utente di AWS Billing .

È possibile aggiungere tag di allocazione `ElastiCache` dei costi a cluster `ElastiCache` progettati autonomamente. Quando aggiungi, elenchi, modifichi, copi o rimuovi un tag, l'operazione viene applicata solo al cluster specificato.

### Caratteristiche dei tag di allocazione dei costi `ElastiCache`

- I tag di allocazione dei costi vengono applicati alle `ElastiCache` risorse specificate in CLI and API operations come. ARN Il tipo di risorsa sarà un cluster.

Esempio ARN: `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 ElastiCache risorsa 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 alcun significato semantico ai tag. I tag vengono interpretati rigorosamente come stringhe di caratteri. AWS non imposta automaticamente alcun tag su nessuna ElastiCache risorsa.

## Gestione dei tag di allocazione dei costi utilizzando il AWS CLI

È possibile utilizzare i AWS CLI per aggiungere, modificare o rimuovere i tag di allocazione dei costi.

I tag di allocazione dei costi vengono applicati ai cluster. ElastiCache Il cluster da etichettare viene specificato utilizzando un ARN (Amazon Resource Name).

Arn di esempio `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

### Argomenti

- [Elencare i tag utilizzando il AWS CLI](#)
- [Aggiungere tag utilizzando il AWS CLI](#)
- [Modifica dei tag utilizzando il AWS CLI](#)
- [Rimuovere i tag utilizzando il AWS CLI](#)

## Elencare i tag utilizzando il AWS CLI

È possibile utilizzare il AWS CLI per elencare i tag su una ElastiCache risorsa esistente utilizzando il [list-tags-for-resource](#) operazione.

Il codice seguente utilizza AWS CLI per elencare i tag sul cluster Memcached `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
```

Per Windows:

```
aws elasticache list-tags-for-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Il codice seguente utilizza AWS CLI per elencare i tag sul nodo Valkey o Redis `my-cluster-001` nel OSS `my-cluster 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 ci sono tag sulla risorsa, l'output sarà vuoto. TagList

```
{
```

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

Per ulteriori informazioni, consulta AWS CLI il ElastiCache [list-tags-for-resource](#).

## Aggiungere tag utilizzando il AWS CLI

È possibile utilizzare il AWS CLI per aggiungere tag a una ElastiCache risorsa esistente utilizzando il [add-tags-to-resource](#) CLI operazione. Se la nuova chiave di tag non esiste sulla risorsa, la chiave e il valore vengono aggiunti alla risorsa. Se la chiave esiste già sulla risorsa, il valore associato a quella chiave viene aggiornato al nuovo valore.

Il codice seguente utilizza AWS CLI per aggiungere le chiavi Service e Region con i valori elasticache e us-west-2 rispettivamente al nodo my-cluster-001 nel cluster my-cluster nella regione us-west-2.

### Memcached

Per Linux, macOS o Unix:

```
aws elasticache add-tags-to-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --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 ^
 --tags Key=Service,Value=elasticache ^
 Key=Region,Value=us-west-2
```

### Redis

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, consulta il AWS CLI ElastiCache [add-tags-to-resource](#).

È inoltre possibile utilizzare AWS CLI per aggiungere tag a un cluster quando si crea un nuovo cluster utilizzando l'operazione [create-cache-cluster](#). Non è possibile aggiungere tag durante la creazione di un cluster utilizzando la console di ElastiCache gestione. Dopo aver creato il cluster, puoi quindi utilizzare la console per aggiungere i tag al cluster.

## Modifica dei tag utilizzando il AWS CLI

È possibile utilizzare il AWS CLI per modificare i tag su un ElastiCache cluster.

Per modificare i tag:

- Utilizzo [add-tags-to-resource](#) per aggiungere un nuovo tag e valore o per modificare il valore associato a un tag esistente.
- Utilizzo [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.

## Rimuovere i tag utilizzando il AWS CLI

È possibile utilizzare AWS CLI per rimuovere i tag da un cluster esistente ElastiCache (Memcached) utilizzando il [remove-tags-from-resource](#) operazione.

Per Memcached, il codice seguente utilizza 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 \
 --tag-keys PM Service
```

Per Windows:

```
aws elasticache remove-tags-from-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
 --tag-keys PM Service
```

Per Redis, il codice seguente utilizza 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, consulta il AWS CLI ElastiCache [remove-tags-from-resource](#).

## Gestione dei tag di allocazione dei costi utilizzando ElastiCache API

È possibile utilizzare i ElastiCache API per aggiungere, modificare o rimuovere i tag di allocazione dei costi.

I tag di allocazione dei costi vengono applicati ai cluster ElastiCache Memcached. Il cluster da etichettare viene specificato utilizzando un ARN (Amazon Resource Name).

Arn di esempio: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

### Argomenti

- [Elencare i tag utilizzando il ElastiCache API](#)
- [Aggiungere tag utilizzando il ElastiCache API](#)
- [Modificare i tag utilizzando il ElastiCache API](#)
- [Rimuovere i tag utilizzando il ElastiCache API](#)

### Elencare i tag utilizzando il ElastiCache API

È possibile utilizzare il ElastiCache API per elencare i tag su una risorsa esistente utilizzando il [ListTagsForResource](#) operazione.

Per Memcached, il codice seguente utilizza ElastiCache API per elencare i tag sulla risorsa `my-cluster` nella regione `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListTagsForResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Per Redis, il codice seguente utilizza ElastiCache API per elencare i tag sulla risorsa `my-cluster-001` nella regione `us-west-2`.

```
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>
```

## Aggiungere tag utilizzando il ElastiCache API

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

Il codice seguente utilizza us-west-2 rispettivamente ElastiCache API per aggiungere le chiavi Service e Region con i valori elasticache. Per Memcached, questo viene applicato alla risorsa my-cluster Per Redis, questo viene applicato alla risorsa my-cluster-001 nella regione us-west-2.

### Memcached

```
https://elasticache.us-west-2.amazonaws.com/
?Action=AddTagsToResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
&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>
```

### Redis

```
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 la pagina [AddTagsToResource](#) nell'Amazon ElastiCache API Reference.

## Modificare i tag utilizzando il ElastiCache API

È possibile utilizzare il ElastiCache API per modificare i tag su un ElastiCache cluster.

Per modificare il valore di un tag:

- Utilizzo [AddTagsToResource](#) operazione per aggiungere un nuovo tag e valore o per modificare il valore di un tag esistente.
- Utilizzo [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.

Utilizzo [RemoveTagsFromResource](#) per rimuovere i tag dalla risorsa.

## Rimuovere i tag utilizzando il ElastiCache API

È possibile utilizzare ElastiCache API per rimuovere i tag da un cluster esistente ElastiCache (Memcached) utilizzando il [RemoveTagsFromResource](#) operazione.

Il codice seguente utilizza ElastiCache API per rimuovere i tag con le chiavi `Service` e `Region` dal nodo `my-cluster-001` nel cluster `my-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 dell'obiettivo Amazon ElastiCache Well-Architected

Questa sezione descrive Amazon ElastiCache Well-Architected Lens, una raccolta di principi di progettazione e linee guida per la progettazione di carichi di lavoro ben architettati. ElastiCache

- The ElastiCache Lens è complementare al [AWS Well-Architected](#) Framework.
- Ogni pilastro ha una serie di domande per aiutare a iniziare 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 AWS risorse 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 set di componenti che insieme 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.

### Note

Questa guida non è stata aggiornata per includere informazioni sulla memorizzazione nella cache ElastiCache serverless e sul nuovo motore Valkey.

### Argomenti

- [Pilastro dell'eccellenza operativa di Amazon ElastiCache Well-Architected Lens](#)

- [Pilastro della sicurezza delle ElastiCache lenti Amazon Well-Architected](#)
- [Pilastro dell'affidabilità delle ElastiCache lenti Amazon Well-Architected](#)
- [Pilastro dell'efficienza delle prestazioni delle lenti Amazon ElastiCache Well-Architected](#)
- [Pilastro dell'ottimizzazione dei costi delle lenti Amazon ElastiCache Well-Architected](#)

## 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

- [OE 1: Come comprendi e rispondi agli avvisi e agli eventi generati dal tuo cluster? ElastiCache](#)
- [OE 2: Quando e come ridimensionate i cluster esistenti? ElastiCache](#)
- [OE 3: Come si gestiscono le ElastiCache risorse e la manutenzione del cluster up-to-date?](#)
- [OE 4: Come gestite le connessioni dei clienti ai vostri ElastiCache cluster?](#)
- [OE 5: Come si distribuiscono i ElastiCache componenti per un carico di lavoro?](#)
- [EO 6: come si pianificano e si contengono gli errori?](#)
- [OE 7: Come si risolvono gli eventi del motore Valkey o RedisOSS?](#)

### OE 1: Come comprendi e rispondi agli avvisi e agli eventi generati dal tuo cluster? ElastiCache

Introduzione a livello di domanda: quando gestisci ElastiCache i cluster, puoi facoltativamente ricevere notifiche e avvisi 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 scalabilità, 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] Controlla gli eventi generati da ElastiCache sulla ElastiCache console (dopo aver selezionato la tua regione) o utilizzando il [comando Amazon Command Line Interface \(AWS CLI\) describe-events](#) e il [ElastiCache API](#) Configura ElastiCache l'invio di notifiche per importanti eventi del cluster utilizzando Amazon Simple Notification Service (AmazonSNS). L'utilizzo di Amazon SNS con i tuoi 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 scalabilità, failover, riavvio del nodo, istantanea creata, modifica dei parametri del cluster, rinnovo del certificato CA, eventi di errore (errore di provisioning del cluster - VPC o ENI -, errori di scalabilità - - ed errori di snapshot). ENI 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
  - ElastiCacheOSS: (solo Valkey o Redis) SnapshotFailed
- [Risorse]:
  - [Gestione delle SNS notifiche ElastiCache Amazon](#)
  - [Notifiche di eventi e Amazon SNS](#)
- [Ideale] Per automatizzare le risposte agli eventi, sfrutta le funzionalità di AWS prodotti e servizi come SNS Lambda Functions. Segui le best practice apportando modifiche piccole, frequenti e reversibili, come codice per migliorare le tue operazioni nel tempo. È necessario utilizzare i CloudWatch parametri di Amazon per monitorare i cluster.

[Risorse]: [Monitor ElastiCache \(RedisOSS\) \(modalità cluster disabilitata\) legge gli endpoint di replica utilizzando AWS Lambda, Amazon Route 53 e Amazon SNS](#) per un caso d'uso che utilizza Lambda e SNS

## OE 2: Quando e come ridimensionate i cluster esistenti? ElastiCache

Introduzione a livello di domanda: il corretto dimensionamento del ElastiCache cluster 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 rimediare all'utilizzo eccessivo o insufficiente delle risorse, è possibile scalare in due dimensioni. ElastiCache È possibile dimensionare verticalmente aumentando o diminuendo la capacità del nodo. Puoi anche dimensionare orizzontalmente aggiungendo e rimuovendo nodi.

- [Obbligatorio] CPU e l'eccessivo utilizzo della rete sui nodi primari devono essere risolti trasferendo 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. Questo può essere configurato nella libreria OSS client Valkey o Redis connettendosi all'endpoint del ElastiCache lettore per disabilitare la modalità cluster o utilizzando il comando per abilitare la modalità cluster. READONLY

[Risorse]:

- [Ricerca degli endpoint di connessione in ElastiCache](#)
- [Cluster Right-Sizing](#)
- [READONLYComando](#)
- [Obbligatorio] Monitora l'utilizzo di 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 (RedisOSS) è disattivata, i nodi primari e di replica possono essere scalati 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]:

- [Monitoraggio delle best practice con ElastiCache \(RedisOSS\) tramite Amazon CloudWatch](#)
- [Scalabilità dei cluster ElastiCache \(RedisOSS\)](#)
- [Scalabilità ElastiCache per 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 CloudWatch metriche per scansionare intervalli di

tempo più lunghi. Gli insegnamenti derivanti dall'osservazione di lunghi periodi di CloudWatch metriche dovrebbero contribuire alla previsione sull'utilizzo delle risorse del cluster. CloudWatch i punti dati e le metriche sono disponibili per un massimo di 455 giorni.

[Risorse]:

- [Monitoraggio ElastiCache \(RedisOSS\) con metriche CloudWatch](#)
  - [Monitoraggio di Memcached con metriche CloudWatch](#)
  - [Monitoraggio delle best practice con ElastiCache \(RedisOSS\) tramite Amazon CloudWatch](#)
- [Migliore] Se le ElastiCache risorse sono create con l'ausilio di strumenti, è CloudFormation consigliabile apportare modifiche utilizzando CloudFormation modelli per preservare la coerenza operativa ed evitare modifiche non gestite alla configurazione e derive dello stack.

[Risorse]:

- [ElastiCache riferimento al tipo di risorsa per CloudFormation](#)
- [Ideale] Automatizza le operazioni di scalabilità utilizzando i dati operativi del cluster e definisci le soglie per configurare gli allarmi. CloudWatch Usa CloudWatch Events and Simple Notification Service (SNS) per attivare le funzioni Lambda ed eseguirne una ElastiCache API per scalare 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]:

- [Utilizzo di Amazon CloudWatch Alarms](#)
- [Cosa sono CloudWatch gli eventi Amazon?](#)
- [Utilizzo AWS Lambda con Amazon Simple Notification Service](#)
- [ElastiCacheAPIRiferimento](#)

### OE 3: Come si gestiscono le ElastiCache risorse e la manutenzione del cluster up-to-date?

Introduzione a livello di 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 applicative, è necessario creare una simmetria tra le versioni del cluster in tutti i tipi di ElastiCache ambiente: 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. Lo sfruttamento delle ElastiCache funzionalità native consente al team di progettazione di concentrarsi sul raggiungimento degli obiettivi aziendali e non sulla ElastiCache manutenzione.

- [Ottimale] Esegui l'ultima versione del motore disponibile e applica gli aggiornamenti self-service non appena sono disponibili. 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 self-service in Amazon ElastiCache](#)
- [Pagina di aiuto Amazon ElastiCache Managed Maintenance and Service Updates](#)
- [Ottimale] Organizza ElastiCache le tue risorse 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 riferimento al tipo di risorsa per CloudFormation](#)
- [Gruppi di parametri](#)

## OE 4: Come gestite le connessioni dei clienti ai vostri ElastiCache cluster?

Introduzione a livello di domanda: quando si opera su larga scala, è necessario comprendere in che modo i clienti si connettono al ElastiCache cluster per gestire gli aspetti operativi dell'applicazione (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 perderai la capacità di leggere una chiave subito dopo averla scritta a causa della natura asincrona della replica Valkey e Redis. OSS Il WAIT comando può essere sfruttato per migliorare la sicurezza dei dati nel mondo reale e forzare le repliche a confermare le scritture prima di rispondere ai clienti, a un costo complessivo in termini di prestazioni. L'utilizzo dei nodi di replica per le operazioni di lettura può essere configurato nella libreria client ElastiCache (RedisOSS) utilizzando l'endpoint di ElastiCache lettura per la modalità cluster disattivata. Per abilitare la modalità cluster, utilizzate il comando ElastiCache (RedisOSS). READONLY Per molte delle librerie client ElastiCache (RedisOSS), ElastiCache (RedisOSS) READONLY è implementato di default o tramite un'impostazione di configurazione.

[Risorse]:

- [Ricerca degli endpoint di connessione in ElastiCache](#)
- [READONLY](#)
- [Obbligatorio] Usa il pool di connessioni. Stabilire una TCP connessione ha un costo in termini di CPU tempo sia sul lato client che sul lato server e il pooling 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. È possibile implementare il pool di connessioni tramite la libreria client ElastiCache (RedisOSS) (se supportata), con un Framework disponibile per l'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 CPU l'utilizzo necessari per aprire e chiudere le connessioni ed eseguire una stretta di TLS mano se TLS è abilitata nel cluster.

[Risorse]: [Configura ElastiCache \(Redis OSS\)](#) per una maggiore disponibilità

- [Positivo] L'utilizzo delle pipeline (quando i casi d'uso lo consentono) può aumentare significativamente le prestazioni.
  - Con il pipelining riduci il Round-Trip Time (RTT) tra i client dell'applicazione 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](#)

## OE 5: Come si distribuiscono i ElastiCache componenti per un carico di lavoro?

Introduzione a livello di domanda: ElastiCache gli ambienti possono essere distribuiti manualmente tramite la AWS console o programmaticamente tramite toolkit, ecc. APIs CLI Le best practice dell'eccellenza operativa suggeriscono di automatizzare le implementazioni tramite il codice ogni volta che è possibile. Inoltre, ElastiCache i cluster possono essere isolati in base al carico di lavoro o combinati per ottimizzare i costi.

Vantaggio a livello di domanda: la scelta del meccanismo di implementazione più appropriato per i propri ElastiCache ambienti può migliorare Operation Excellence 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 ElastiCache ambienti dedicati per carico di lavoro o combinare più carichi di lavoro in singoli cluster o combinazioni di essi. Comprendere i compromessi può aiutare a trovare un equilibrio tra eccellenza operativa e ottimizzazione dei costi.

- [Obbligatorio] Comprendi le opzioni di implementazione disponibili e automatizza queste procedure quando possibile. ElastiCache Le possibili vie di automazione includono CloudFormationSDK, AWS CLI/e. APIs

[Risorse]:

- [Riferimento al tipo di ElastiCache risorsa Amazon](#)
- [elasticache](#)

- [ElastiCache API Riferimento Amazon](#)
- [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, dimensionamento, scalabilità e gestione delle ElastiCache risorse 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 a livello di domanda: l'eccellenza operativa include l'anticipazione dei guasti eseguendo regolarmente esercizi «pre-mortem» per identificare le potenziali fonti di guasto in modo che possano essere rimosse o mitigate. ElastiCache offre un failover API che consente di simulare eventi di guasto 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](#)

## OE 7: Come si risolvono gli eventi del motore Valkey o RedisOSS?

Introduzione a livello di domanda: l'eccellenza operativa richiede la capacità di esaminare le informazioni a livello di servizio e a livello di motore per analizzare lo stato e lo stato dei cluster. ElastiCache può emettere log del OSS motore Valkey o Redis sia su Amazon che su Amazon Kinesis Data CloudWatch Firehose.

Vantaggio a livello di domanda: l'abilitazione dei log del OSS motore Valkey o Redis sui cluster fornisce informazioni sugli eventi che influiscono sullo stato e sulle prestazioni ElastiCache dei cluster. I log OSS del motore Valkey o Redis forniscono dati direttamente dal motore che non sono disponibili tramite il meccanismo degli eventi. ElastiCache Attraverso un'attenta osservazione sia degli ElastiCache eventi (vedere la precedente OE-1) che dei registri del motore, è possibile determinare un ordine degli eventi durante la risoluzione dei problemi sia dal punto di vista del servizio che dal punto di vista del motore. ElastiCache

- [Obbligatorio] Assicuratevi che la funzionalità di registrazione OSS del motore Redis sia abilitata, disponibile a partire da ElastiCache (Redis) 6.2 e versioni successive. OSS Questa operazione può essere eseguita durante la creazione del cluster o modificando il cluster dopo la creazione.
- Determina se Amazon CloudWatch Logs o Amazon Kinesis Data Firehose è la destinazione appropriata per i log del motore Redis. OSS
- Seleziona un log di destinazione appropriato all'interno di uno CloudWatch o di Kinesis Data Firehose per rendere permanenti 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 di registrazione: [Amazon CloudWatch](#) Logs
- Introduzione ad Amazon CloudWatch Logs: [cos'è Amazon CloudWatch Logs?](#)
- Introduzione ad Amazon Kinesis Data Firehose: [What Is Amazon Kinesis Data Firehose?](#)
- [Migliore] Se utilizzi Amazon CloudWatch Logs, valuta la possibilità di sfruttare Amazon CloudWatch Logs Insights per interrogare i log del OSS motore Valkey o Redis per ottenere informazioni importanti.

Ad esempio, crea una query sul gruppo CloudWatch Log che contiene i log del OSS motore Valkey o Redis che restituiranno eventi contrassegnati da ", ad esempio: LogLevel WARNING

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Risorse]: [analisi dei dati di registro](#) con Logs Insights CloudWatch

## Pilastro della sicurezza delle ElastiCache lenti Amazon Well-Architected

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

- [SEC1: Quali misure state adottando per controllare l'accesso autorizzato ai dati? ElastiCache](#)

- [SEC2: Le vostre applicazioni richiedono un'autorizzazione aggiuntiva ElastiCache rispetto ai controlli basati sulla rete?](#)
- [SEC3: Esiste il rischio che i comandi possano essere eseguiti inavvertitamente, causando la perdita o il fallimento dei dati?](#)
- [SEC4: Come si garantisce la crittografia dei dati inattivi con ElastiCache](#)
- [SEC5: Come si crittografano i dati in transito? ElastiCache](#)
- [SEC6: Come si limita l'accesso alle risorse del piano di controllo?](#)
- [SEC7: Come rilevi e rispondi agli eventi di sicurezza?](#)

## SEC1: Quali misure state adottando per controllare l'accesso autorizzato ai dati? ElastiCache

Introduzione a livello di domanda: tutti i ElastiCache cluster sono progettati per essere accessibili da istanze di Amazon Elastic Compute Cloud in funzioni serverless () o contenitori (AWS Lambda Amazon Elastic Container Service). VPC Lo scenario più comune consiste nell'accedere a un ElastiCache cluster da un'istanza Amazon Elastic Compute Cloud all'interno dello stesso Amazon Virtual Private Cloud (Amazon Virtual Private Cloud). Prima di poterti connettere a un cluster da un'EC2istanza Amazon, devi autorizzare l'EC2istanza Amazon ad accedere al cluster. Per accedere a un ElastiCache cluster in esecuzione in aVPC, è necessario concedere l'accesso alla rete al cluster.

Vantaggio a livello di domanda: l'ingresso della rete nel cluster è controllato tramite gruppi di sicurezza. VPC Un gruppo di sicurezza funge da firewall virtuale per le tue EC2 istanze Amazon per controllare il traffico in entrata e 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 l'implementazione esclusivamente su sottoreti private in modo che siano accessibili solo tramite la rete privata della rete. 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 avrà alcuna regola in entrata definita e, quindi, nessun percorso di ingresso. ElastiCache Per abilitare questa funzionalità, configura una regola in entrata sul gruppo di sicurezza specificando l'indirizzo/intervallo IP di origine, TCP digita il traffico e la porta per il ElastiCache cluster (ad esempio la porta predefinita 6379 per (Redis)). ElastiCache OSS Sebbene sia possibile consentire un set molto ampio di fonti 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 autorizzando solo l'accesso in entrata ai client Valkey o Redis in OSS esecuzione su istanze Amazon Amazon associate a un gruppo di sicurezza specifico. EC2

[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]AWS Identity and Access Management è possibile assegnare politiche a funzioni che consentono loro di accedere ai dati. AWS Lambda ElastiCache Per abilitare questa funzionalità, crea un ruolo di IAM esecuzione con l'AWSLambdaVPCAccessExecutionRole autorizzazione, quindi assegna il ruolo alla AWS Lambda funzione.

[Risorse]: Configurazione di una funzione Lambda per accedere ad Amazon in ElastiCache VPC Amazon: [Tutorial: Configurazione di una funzione Lambda per accedere ad ElastiCache](#) Amazon in Amazon VPC

## SEC2: Le vostre applicazioni richiedono un'autorizzazione aggiuntiva ElastiCache rispetto ai controlli basati sulla rete?

Introduzione a livello di domanda: negli scenari in cui è necessario limitare o controllare l'accesso ai cluster ElastiCache (RedisOSS) a livello di singolo client, si consiglia di effettuare l'autenticazione tramite il comando (Redis). ElastiCache OSS AUTH ElastiCache I token di autenticazione (RedisOSS), con gestione opzionale di utenti e gruppi di utenti, consentono a ElastiCache (RedisOSS) di richiedere una password prima di consentire ai client di eseguire comandi e chiavi di accesso, migliorando così la sicurezza del piano dati.

Vantaggio a livello di domanda: per contribuire a proteggere i dati, ElastiCache (RedisOSS) fornisce meccanismi di protezione contro l'accesso non autorizzato ai dati. Ciò include l'applicazione del Role-Based Access Control (RBAC) AUTH o del AUTH token (password) a cui i client devono connettersi prima di eseguire comandi autorizzati. ElastiCache

- [Ideale] Per ElastiCache (RedisOSS) 6.x e versioni successive, definisci 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 utilizzarloRBAC, deve essere selezionato

al momento della creazione del cluster e la crittografia in transito deve essere abilitata. Assicurati di utilizzare un OSS client Valkey o Redis che supporti TLS per poterlo sfruttare. RBAC

[Risorse]:

- [Richiesta RBAC di iscrizione a un gruppo di replica per \(Redis\) ElastiCache OSS](#)
- [Specifica delle autorizzazioni mediante una stringa di accesso](#)
- [ACL](#)
- [Versioni supportate ElastiCache \(RedisOSS\)](#)
- [Ideale] Per le versioni ElastiCache (RedisOSS) precedenti alla 6.x, oltre a impostare token/password complessi e mantenere una politica rigorosa in materia di password per ElastiCache (RedisOSS)AUTH, è consigliabile ruotare la password/il token. ElastiCache può gestire fino a due (2) token di autenticazione in un dato momento. Puoi anche modificare il cluster per richiedere esplicitamente l'uso di token di autenticazione.

[Risorse]: [modifica del AUTH token su un cluster esistente ElastiCache \(RedisOSS\)](#)

SEC3: Esiste il rischio che i comandi possano essere eseguiti inavvertitamente, causando la perdita o il fallimento dei dati?

Introduzione a livello di domanda: Esistono diversi OSS comandi Valkey o Redis che possono avere un impatto negativo sulle operazioni se eseguiti per errore o da attori malintenzionati. Questi comandi possono avere conseguenze impreviste dal punto di vista delle prestazioni e della sicurezza dei dati. Ad esempio, uno sviluppatore può richiamare regolarmente il FLUSHALL comando in un ambiente di sviluppo e, a causa di un errore, può tentare inavvertitamente di richiamare questo comando su un sistema di produzione, con conseguente perdita accidentale di dati.

Vantaggio a livello di domanda: a partire da ElastiCache (RedisOSS) 5.0.3, è possibile rinominare determinati comandi che potrebbero compromettere il carico di lavoro. La ridenominazione dei comandi può aiutare a evitare che vengano eseguiti inavvertitamente sul cluster.

- [Obbligatorio]

[Risorse]:

- [ElastiCache \(Redis\) versione 5.0.3 \(obsoleta, usa la versione OSS 5.0.6\)](#)
- [Modifiche ai parametri OSS Redis 5.0.3](#)
- [Sicurezza Redis OSS](#)

## SEC4: Come si garantisce la crittografia dei dati inattivi con ElastiCache

Introduzione a livello di domanda: sebbene ElastiCache (RedisOSS) sia un archivio dati in memoria, è possibile crittografare qualsiasi dato che possa essere reso persistente (in 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 a livello di domanda: ElastiCache (RedisOSS) offre una crittografia opzionale a riposo per aumentare la sicurezza dei dati.

- [Obbligatorio] La crittografia a riposo può essere abilitata su un ElastiCache cluster (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 fornirà e gestirà le chiavi utilizzate nella crittografia at-rest.

[Risorse]:

- [Vincoli di crittografia At-Rest](#)
- [Abilitazione della crittografia dei dati inattivi](#)
- [Ideale] Sfrutta i tipi di EC2 istanze Amazon 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 ambienti di sicurezza dei dati più rigorosi, AWS Key Management Service (KMS) può essere utilizzato per gestire automaticamente le Customer Master Keys (). Grazie all'integrazione con AWS Key Management Service, puoi creare, possedere e gestire le chiavi utilizzate per la crittografia dei dati inattivi per il tuo cluster ElastiCache (RedisOSS).

[Risorse]:

- [Utilizzo di chiavi gestite dal cliente da AWS Key Management Service](#)
- [AWS Servizio di gestione delle chiavi](#)
- [AWS KMSconcetti](#)

## SEC5: Come si crittografano i dati in transito? 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 i client delle applicazioni e i nodi del cluster. ElastiCache (RedisOSS) 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 a livello di domanda: la crittografia ElastiCache in transito di Amazon è una funzionalità opzionale che consente di aumentare la sicurezza dei 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 (RedisOSS) (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 comprenderne l'impatto, si consiglia di eseguire un benchmark del carico di lavoro prima e dopo l'attivazione. encryption-in-transit

[Risorse]:

- [Panoramica della crittografia dei dati in transito](#)

## SEC6: Come si limita l'accesso alle risorse del piano di controllo?

Introduzione a livello di domanda: IAM le politiche e ARN abilitano controlli di accesso granulari per ElastiCache (RedisOSS), che consentono un controllo più rigoroso della gestione della creazione, della modifica e dell'eliminazione dei ElastiCache cluster (Redis). OSS

Vantaggio a livello di domanda: la gestione ElastiCache delle risorse Amazon, come gruppi di replica, nodi, ecc., può essere limitata agli AWS account che dispongono di autorizzazioni specifiche basate su IAM policy, migliorando la sicurezza e l'affidabilità delle risorse.

- [Obbligatorio] Gestisci l'accesso alle ElastiCache risorse di Amazon assegnando AWS Identity and Access Management policy specifiche agli AWS utenti, permettendo un controllo più preciso su quali account possono eseguire quali azioni sui cluster.

[Risorse]:

- [Panoramica della gestione delle autorizzazioni di accesso alle tue risorse ElastiCache](#)
- [Utilizzo di politiche \(IAMpolitiche\) basate sull'identità per Amazon ElastiCache](#)

## SEC7: Come rilevi e rispondi agli eventi di sicurezza?

Introduzione a livello di domanda: ElastiCache, se distribuito con RBAC enabled, esporta le CloudWatch metriche per notificare agli utenti gli eventi di sicurezza. Queste metriche aiutano a identificare i tentativi falliti di autenticazione, accesso alle chiavi o esecuzione di comandi per i quali gli utenti che effettuano la connessione RBAC non sono autorizzati.

Inoltre, le risorse relative a AWS prodotti e servizi 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 successivi.

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 le metriche pubblicate relative agli errori di autenticazione e autorizzazione CloudWatch . RBAC
  - AuthenticationFailures = Tentativi falliti di autenticazione su Valkey o Redis OSS
  - KeyAuthorizationFailures = Tentativi falliti da parte degli utenti di accedere alle chiavi senza autorizzazione
  - CommandAuthorizationFailures = Tentativi falliti da parte degli utenti di eseguire comandi senza autorizzazione

[Risorse]:

- [Metriche per Valkey o Redis OSS](#)
- [Best practice] Ti consigliamo di configurare avvisi e notifiche su queste metriche e rispondere se necessario.

[Risorse]:

- [Utilizzo degli CloudWatch allarmi Amazon](#)
- [Ottimale] Usa il OSS ACL LOG comando Valkey o Redis per raccogliere ulteriori dettagli

[Risorse]:

- [ACL LOG](#)
- [Migliore] Acquisisci familiarità con le funzionalità dei AWS prodotti e dei servizi per quanto riguarda il monitoraggio, la registrazione e l'analisi delle implementazioni e degli eventi ElastiCache

[Risorse]:

- [Registrazione delle ElastiCache API chiamate Amazon con AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-controllo di backup](#)
- [Monitoraggio dell'uso con Metrics CloudWatch](#)

## Pilastro dell'affidabilità delle ElastiCache lenti Amazon Well-Architected

Il pilastro dell'affidabilità si concentra sui carichi di lavoro che svolgono le funzioni previste e su come recuperare rapidamente in caso di mancato soddisfacimento delle richieste. Gli argomenti chiave includono la progettazione di sistemi distribuiti, la pianificazione del ripristino e l'adattamento ai requisiti in evoluzione.

### Argomenti

- [REL1: In che modo supportate le implementazioni di architetture ad alta disponibilità \(HA\)?](#)
- [REL2: In che modo state raggiungendo i Recovery Point Objectives \(RPOs\)? ElastiCache](#)
- [REL3: In che modo supportate i requisiti di disaster recovery \(DR\)?](#)
- [REL4: Come si pianificano efficacemente i failover?](#)
- [REL5: I vostri ElastiCache componenti sono progettati per essere scalabili?](#)

### REL1: In che modo supportate le implementazioni di architetture ad alta disponibilità (HA)?

Introduzione a livello di domanda: comprendere l'architettura ad alta disponibilità di Amazon ti ElastiCache consentirà di operare in uno stato resiliente durante gli eventi di disponibilità.

Vantaggio a livello di domanda: l'architettura dei ElastiCache cluster in modo che siano resilienti ai guasti garantisce una maggiore disponibilità per le distribuzioni. ElastiCache

- [Obbligatorio] Determina il livello di affidabilità richiesto per il tuo cluster. ElastiCache 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 memorizzazione nella cache: ElastiCache (Memcached) vs ElastiCache (Redis) OSS

1. ElastiCache (Memcached) non fornisce alcun meccanismo di replica e viene utilizzato principalmente per carichi di lavoro temporanei.

## 2. ElastiCache (Redis) offre le funzionalità HA descritte di seguito OSS

- [Ideale] Per carichi di lavoro che richiedono HA, usa ElastiCache (RedisOSS) in modalità cluster con un minimo di due repliche per shard, anche per carichi di lavoro con requisiti di throughput ridotti che richiedono solo uno shard.

### 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 suddivisi, un minimo di tre shard offre un ripristino più rapido durante gli eventi di failover, poiché il Valkey o il Redis OSS Cluster Protocol richiedono 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).

ElastiCache (RedisOSS) ha aggiunto prestazioni ottimizzate su questi nodi. Di conseguenza, si ottengono migliori prestazioni di replica e sincronizzazione, con conseguente maggiore disponibilità complessiva.

### 5. Monitoraggio e dimensioni corrette per far fronte ai picchi di traffico previsti: in caso di carico intenso, il motore ElastiCache (RedisOSS) potrebbe non rispondere, con ripercussioni sulla disponibilità. BytesUsedForCache e DatabaseMemoryUsagePercentage sono buoni indicatori dell'utilizzo della memoria, mentre ReplicationLag sono un indicatore dello stato della replica in base alla velocità di scrittura. Puoi utilizzare queste metriche per attivare il dimensionamento dei cluster.

### 6. Garantite la resilienza lato client eseguendo test con il [failover API prima di un evento di failover di produzione](#).

[Risorse]:

- [Configura ElastiCache \(Redis\) per una maggiore disponibilità OSS](#)
- [Alta disponibilità utilizzando gruppi di replica](#)

## REL2: In che modo state raggiungendo i Recovery Point Objectives (RPOs)? ElastiCache

Introduzione a livello di domanda: Comprendi il carico di lavoro RPO per prendere decisioni informate sulle strategie di ElastiCache backup e ripristino.

Vantaggio a livello di domanda: disporre di una RPO strategia in atto può migliorare la continuità aziendale in caso di scenari di disaster recovery. La progettazione delle politiche di backup e ripristino può aiutarti a raggiungere gli obiettivi dei punti di ripristino (RPO) per i tuoi dati. ElastiCache ElastiCache (RedisOSS) offre funzionalità di snapshot archiviate in Amazon S3, insieme a una politica 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 le RPO tue implementazioni. ElastiCache
  - Tieni presente che Memcached non offre processi di backup.
  - Esamina le funzionalità delle funzionalità di ElastiCache Backup e ripristino.
- [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.
- [Buono] Sfrutta la funzionalità di backup pianificato ElastiCache per eseguire regolarmente il backup dei dati durante una finestra definita.
  - Esegui periodicamente il test del ripristino dei tuoi backup.
- [Risorse]:
  - [Redis OSS](#)
  - [Backup e ripristino per ElastiCache \(RedisOSS\)](#)
  - [Creazione di backup manuali](#)
  - [Pianificazione di backup automatici](#)



- [Cluster di backup e ripristino ElastiCache \(RedisOSS\)](#)

### REL3: In che modo supportate i requisiti di disaster recovery (DR)?

Introduzione a livello di domanda: il disaster recovery è un aspetto importante di qualsiasi pianificazione del carico di lavoro. ElastiCache (RedisOSS) offre diverse opzioni per implementare il disaster recovery in base ai requisiti di resilienza del carico di lavoro. Con Amazon ElastiCache Global Datastore, puoi scrivere sul tuo cluster ElastiCache (RedisOSS) in una regione e avere i dati disponibili per essere letti da altri due cluster di replica interregionali, abilitando così letture a bassa latenza e disaster recovery 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 DR per tutti i componenti in base ai requisiti del carico di lavoro. ElastiCache ElastiCache è unico in quanto alcuni casi d'uso sono completamente effimeri e non richiedono alcuna strategia di DR, mentre altri si collocano all'estremità opposta e richiedono una strategia di DR estremamente solida. 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 Cluster-Mode abilitata nelle architetture Multi-AZ, con un minimo di 3 disponibili. AZs
- 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 utilizzando datastore globali AWS](#)

- [Ripristino da un backup con ridimensionamento cluster opzionale](#)
- [Riduzione al minimo dei tempi di inattività in ElastiCache \(Redis\) con Multi-AZ OSS](#)

## REL4: Come si pianificano efficacemente i failover?

Introduzione a livello di domanda: abilitare Multi-AZ con failover automatici è una best practice. ElastiCache In alcuni casi, ElastiCache (RedisOSS) sostituisce i nodi primari nell'ambito 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. Il successo dei failover dipende sia dalla configurazione della libreria client sia dalla configurazione della ElastiCache libreria client.

Vantaggio a livello di domanda: seguire le migliori pratiche per i ElastiCache failover in combinazione con la libreria client specifica ElastiCache (RedisOSS) 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. Ciò viene spesso ottenuto mediante le impostazioni di configurazione nella libreria client ElastiCache (RedisOSS), 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 al minimo dei tempi di inattività in ElastiCache \(Redis\) con Multi-AZ OSS](#)
- Consulta le best practice della tua libreria client ElastiCache (RedisOSS).
- [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 Valkey e Redis, nonché le CloudWatch metriche disponibili per segnalare il ritardo di OSS replica tra i nodi primari e di replica. Per i casi d'uso che richiedono una maggiore sicurezza dei dati, sfrutta il WAIT comando per forzare le repliche a confermare le scritture prima di rispondere ai client connessi.

[Risorse]:

- [Metriche per Valkey o Redis OSS](#)
- [Monitoraggio delle best practice con ElastiCache \(RedisOSS\) tramite Amazon CloudWatch](#)

- [Migliore] Convalida regolarmente la reattività dell'applicazione durante il failover utilizzando il Test Failover. ElastiCache API

[Risorse]:

- [Test del failover automatico su una replica di lettura su Amazon ElastiCache \(Redis\) OSS](#)
- [Test del failover automatico](#)

## REL5: I vostri ElastiCache componenti sono progettati per essere scalabili?

Introduzione a livello di domanda: comprendendo le capacità di scalabilità e le topologie di implementazione disponibili, ElastiCache i componenti possono adattarsi nel tempo per soddisfare i mutevoli requisiti del carico di lavoro. ElastiCache offre una scalabilità a 4 vie: entrata/uscita (orizzontale) e su/giù (verticale).

Vantaggi a livello di domanda: seguire le migliori pratiche per le ElastiCache implementazioni offre la massima flessibilità di scalabilità, oltre a soddisfare il principio Well Architected di scalabilità orizzontale per ridurre al minimo l'impatto dei guasti.

- [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 informazioni READIOPS: aggiungi repliche
  - Per ulteriori informazioni WRITEOPS: aggiungi frammenti (scalabilità orizzontale)
  - Per ulteriori I/O di rete: utilizza istanze ottimizzate per la rete (dimensionamento verticale)
- [Ideale] Implementa i ElastiCache componenti con la modalità Cluster abilitata, con una preferenza verso un numero maggiore di nodi più piccoli anziché un numero inferiore di nodi più 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
- [Buono] Se la modalità cluster è disattivata, sfrutta le repliche di lettura per aumentare la capacità di lettura complessiva. ElastiCache supporta fino a 5 repliche di lettura in modalità cluster disattivata, oltre al ridimensionamento verticale.
- [Risorse]:

- [Scalabilità ElastiCache dei cluster \(Redis\) OSS](#)
- [Dimensionamento verso l'alto online](#)
- [Scalabilità ElastiCache per cluster Memcached](#)

## Pilastro dell'efficienza delle prestazioni delle lenti Amazon ElastiCache Well-Architected

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

- [PE 1: Come monitorate le prestazioni del vostro ElastiCache cluster Amazon?](#)
- [PE 2: Come state distribuendo il lavoro tra i ElastiCache nodi del cluster?](#)
- [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?](#)
- [PE 6: In che modo modellate e interagite con i dati ElastiCache?](#)
- [PE 7: Come si registrano i comandi a esecuzione lenta nel ElastiCache cluster Amazon?](#)
- [PE8: In che modo l'Auto Scaling aiuta ad aumentare le prestazioni del ElastiCache cluster?](#)

### PE 1: Come monitorate le prestazioni del vostro ElastiCache cluster Amazon?

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 CloudWatch metriche durante l'esecuzione di questi test per comprendere le metriche disponibili e stabilire una base di riferimento delle prestazioni.
- [Ideale] Per i carichi di lavoro ElastiCache (RedisOSS), rinomina i comandi computazionalmente costosi, ad esempio per limitare la capacità degli utenti di eseguire comandi di blocco sui cluster di produzione. KEYS
  - ElastiCache I carichi di lavoro (RedisOSS) che eseguono il motore 6.x possono sfruttare il controllo degli accessi basato sui ruoli per limitare determinati comandi. L'accesso ai comandi può essere controllato creando utenti e gruppi di utenti con la AWS console oppure CLI associando i gruppi di utenti a un cluster (Redis). ElastiCache OSS In Redis OSS 6, quando RBAC è abilitato, possiamo usare «- @dangerous" e non consentirà comandi costosi come KEYS, MONITORSORT, ecc. per quell'utente.
  - Per la versione 5.x del motore, rinomina i comandi utilizzando il `rename-commands` parametro nel gruppo di parametri del cluster ElastiCache (RedisOSS).
- [Consigliato] Analizza le query lente ed esamina le tecniche di ottimizzazione.
  - Per i carichi di lavoro ElastiCache (RedisOSS), scopri di più sulle tue query analizzando lo Slow Log. Ad esempio, puoi utilizzare il seguente comando `valkey-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 strutture di dati complesse ElastiCache (Redis). OSS 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 ElastiCache (RedisOSS), `redis-benchmark` fornisce un'interfaccia semplice per testare le prestazioni di diversi comandi utilizzando 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'uso con Metrics CloudWatch](#)
  - [Utilizzo degli CloudWatch allarmi Amazon](#)
  - [Parametri specifici di Valkey e Redis OSS](#)

- [SLOWLOG](#)
- [benchmark](#)

## PE 2: Come state distribuendo il lavoro tra i ElastiCache nodi del cluster?

Introduzione a livello di domanda: il modo in cui l'applicazione si connette ElastiCache ai nodi Amazon 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] Consenti ai client di connettersi all'endpoint corretto. ElastiCache
  - ElastiCache (RedisOSS) implementa diversi endpoint in base alla modalità cluster in uso. Se la modalità cluster è abilitata, ElastiCache fornirà un endpoint di configurazione. Per la modalità cluster disattivata, 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 multinodo, fornisce un endpoint di configurazione che abilita l'Auto Discovery ElastiCache . 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](#).
- [Migliore] Sfrutta la modalità cluster ElastiCache (RedisOSS) abilitata per migliorare la scalabilità.
  - ElastiCache I cluster (RedisOSS) (modalità cluster abilitata) supportano [operazioni di scalabilità online](#) (out/in e up/down) per aiutare a distribuire i dati in modo dinamico tra gli shard. 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 hashslot tra gli shard disponibili nel cluster (Redis) (abilitato in modalità cluster). ElastiCache OSS 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.

- Considerate l'impatto delle strutture di OSS dati multidimensionali di Valkey o Redis come elenchi, stream, set, ecc. Queste strutture di dati sono archiviate in singole chiavi, che risiedono su un singolo 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. Per i carichi di lavoro ElastiCache (RedisOSS), puoi rilevare i tasti di scelta rapida utilizzando `valkey-cli --hotkeys` se è in atto una politica di LFU massima memoria.
- Prendi in considerazione la replica delle chiavi hot su più nodi per distribuire l'accesso in modo più uniforme. Questo approccio richiede che il client scriva su più nodi primari (lo stesso OSS nodo Valkey o Redis non fornirà questa funzionalità) e mantenga un elenco di nomi di chiavi da cui leggere, oltre al nome della chiave originale.
- ElastiCache [con Valkey 7.2 e versioni successive e Redis OSS versione 6 e successive supportano la memorizzazione nella cache lato client assistita dal server](#). Ciò consente alle applicazioni di attendere le modifiche a una chiave prima di effettuare chiamate di rete a ElastiCache
- [Risorse]:
  - [Configura ElastiCache con Valkey e Redis OSS per una maggiore disponibilità](#)
  - [Ricerca degli endpoint di connessione in ElastiCache](#)
  - [Best practice per il bilanciamento del carico](#)
  - [Resharding online per Valkey o Redis OSS \(modalità cluster abilitata\)](#)
  - [Memorizzazione nella cache lato client in Valkey e Redis OSS](#)

### 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 a livello di domanda: la memorizzazione nella cache è un carico di lavoro comune ElastiCache ed è importante comprendere 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]:

- [Metriche per Valkey e Redis OSS](#)
- [Obbligatorio] Misura e raccogli le prestazioni della cache delle applicazioni insieme ai valori di latenza e CPU utilizzo per capire se è necessario apportare modifiche ai componenti dell'applicazione o ad altri componenti dell'applicazione. time-to-live ElastiCache fornisce un set di CloudWatch metriche per le latenze aggregate per ogni struttura di dati. Queste metriche di latenza vengono calcolate utilizzando la statistica commandstats del comando ElastiCache (RedisOSS) e non includono la rete e il tempo di INFO I/O. Questo è solo il tempo impiegato da ElastiCache (Redis) per elaborare le operazioni. OSS

[Risorse]:

- [Metriche per Valkey e Redis OSS](#)
- [Monitoraggio delle best practice con ElastiCache \(RedisOSS\) tramite 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 per Memcached](#)
- [Caching Best Practices](#)



- [Prestazioni su larga scala con Amazon ElastiCache Whitepaper](#)

EP 4: in che modo il carico di lavoro ottimizza l'uso delle risorse e delle connessioni di rete?

Introduzione a livello di domanda: ElastiCache (RedisOSS) e ElastiCache (Memcached) sono supportati da molti client applicativi 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 della connessione 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]:
  - [ElastiCache \(RedisOSS\) - Datastore globale](#)
  - [Parametri Memcached specifici](#)
  - [ElastiCache \(RedisOSS\) 5.0.3 migliora la gestione dell'I/O per aumentare le prestazioni](#)
  - [Metriche per Valkey e Redis OSS](#)

- [Configura ElastiCache \(RedisOSS\) per una maggiore disponibilità](#)

## EP 5: come si gestisce l'eliminazione e/o l'espulsione delle chiavi?

Introduzione a livello di domanda: i carichi di lavoro hanno requisiti e comportamenti previsti diversi quando un nodo del cluster si avvicina ai limiti di consumo di memoria. ElastiCache (RedisOSS) ha politiche 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 politica può essere gestita utilizzando il gruppo di parametri del cluster `maxmemory-policy` on the ElastiCache (RedisOSS).
  - La politica predefinita `volatile-lru` libera memoria eliminando le chiavi con un tempo di scadenza (valore) impostato. TTL Le politiche utilizzate meno frequentemente (LFU) e quelle utilizzate meno di recente (LRU) rimuovono le chiavi in base all'utilizzo.
  - Per i carichi di lavoro Memcached, esiste una LRU politica predefinita che controlla gli sfratti su ciascun nodo. Il numero di sfratti sul tuo ElastiCache cluster Amazon può essere monitorato utilizzando la metrica Sfratti su 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.
  - Per i carichi di lavoro ElastiCache (RedisOSS), quando si rimuovono esplicitamente le chiavi dal cluster, `UNLINK` è come: rimuove le chiavi specificate. `DEL` 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 ElastiCache (RedisOSS) 6.x, il comportamento del `DEL` comando può essere modificato nel gruppo di parametri utilizzando il parametro. `lazyfree-lazy-user-del`
- [Risorse]:
  - [Configurazione dei parametri del motore utilizzando gruppi di ElastiCache parametri](#)

- [UNLINK](#)
- [Gestione finanziaria nel cloud con AWS](#)

## PE 6: In che modo modellate e interagite con i dati ElastiCache?

Introduzione a livello di domanda: ElastiCache dipende in larga misura dall'applicazione dalle strutture di dati e dal modello di dati utilizzati, ma deve anche considerare l'archivio dati sottostante (se presente). Comprendi le strutture di dati ElastiCache (RedisOSS) disponibili e assicurati di utilizzare le strutture di dati più appropriate per le tue esigenze.

Vantaggio a livello di domanda: la modellazione dei dati 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 ElastiCache (RedisOSS) ha le 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](#)
- I comandi [Best] ElastiCache (RedisOSS) 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. Grazie al design a thread singolo di ElastiCache (RedisOSS), un grande volume di operazioni ad alta complessità temporale si tradurrà in prestazioni inferiori e potenziali timeout operativi.

[Risorse]:

- [Comandi](#)
- [Ideale] Utilizzalo APIs per ottenere GUI visibilità sul modello di dati nel cluster.

[Risorse]:

- [Redis Commander OSS](#)

- [Browser Redis OSS](#)
- [Redsmin](#)

## PE 7: Come si registrano i comandi a esecuzione lenta nel ElastiCache cluster Amazon?

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 determinano prestazioni scadenti e quali comandi che impediscono al motore di funzionare in modo ottimale. ElastiCache (RedisOSS) ha anche la capacità 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 (RedisOSS) utilizza la versione 6.0 o successiva del motore, gruppo di parametri correttamente configurato e SLOWLOG registrazione abilitata sul cluster.
  - I parametri richiesti sono disponibili solo quando la compatibilità della versione del motore è impostata su Valkey 7.2 e versioni successive o Redis versione 6.0 o successiva. OSS
  - SLOWLOG la registrazione si verifica quando il tempo di esecuzione di un comando sul server impiega più tempo di un 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.
- [Migliore] Sfrutta le nostre funzionalità di CloudWatch Kinesis Data Firehose.
  - Utilizza le funzionalità di filtro e allarme di CloudWatch CloudWatch Logs Insights e Amazon Simple Notification Services per monitorare le prestazioni e notificare gli eventi.
  - Utilizza le funzionalità di streaming di Kinesis Data Firehose SLOWLOG per archiviare i log in uno storage permanente o per attivare l'ottimizzazione automatica dei parametri del cluster.
  - Determina se il nostro JSON TEXT formato semplice si adatta meglio alle tue esigenze.
  - Fornisci IAM le autorizzazioni per la pubblicazione su CloudWatch o su 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 all'interno del OSS motore Valkey o 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 limite superiore per il numero di comandi a esecuzione lenta che vengono acquisiti nella memoria Valkey o Redis OSS 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 da `valkey-cli` tramite il comando, valuta la possibilità di aumentare questo valore. SLOWLOG Ciò consente a più comandi di rimanere nella memoria Valkey o Redis. OSS

Se i SLOWLOG dati vengono inviati a CloudWatch Logs o Kinesis Data Firehose, i dati verranno mantenuti e potranno essere analizzati all'esterno del sistema, riducendo ElastiCache la necessità di archiviare un gran numero di comandi a esecuzione lenta nella memoria Valkey o Redis. OSS

- [Risorse]:
  - [Come posso attivare lo slow log in un cluster di cache \(Redis\)? ElastiCache OSS](#)
  - [Consegna dei registri](#)
  - [Parametri specifici di Redis OSS](#)
  - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
  - [Amazon Kinesis Data Firehose](#)

## PE8: In che modo l'Auto Scaling aiuta ad aumentare le prestazioni del ElastiCache cluster?

Introduzione a livello di domanda: implementando la funzionalità di scalabilità OSS automatica Valkey o Redis, ElastiCache i componenti possono adattarsi nel tempo per aumentare o diminuire automaticamente gli shard o le repliche desiderati. Questa operazione può essere eseguita implementando il monitoraggio degli obiettivi o la policy di dimensionamento pianificata.

Vantaggio a livello di domanda: la comprensione e la pianificazione dei picchi del carico di lavoro possono garantire prestazioni di caching migliorate e continuità aziendale. ElastiCache (RedisOSS) Auto Scaling monitora continuamente l'utilizzo di /Memory per assicurarsi che CPU il cluster funzioni ai livelli di prestazioni desiderati.

- [Obbligatorio] Quando si avvia un cluster per (Redis): ElastiCache OSS
  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]:

- [Scalabilità dei cluster in Valkey e Redis OSS \(modalità cluster abilitata\)](#)
- [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. 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.

[Risorse]:

- [Monitoraggio dell'utilizzo con Metrics 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]:

- [Registrare un target scalabile](#)
- [Registrazione di un target scalabile utilizzando il AWS CLI](#)

## Pilastro dell'ottimizzazione dei costi delle lenti Amazon ElastiCache Well-Architected

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

- [COST1: Come identificate e tenete traccia dei costi associati alle vostre risorse? ElastiCache Come si sviluppano meccanismi per consentire agli utenti di creare, gestire ed eliminare le risorse create?](#)
- [COST2: In che modo utilizzate gli strumenti di monitoraggio continuo per ottimizzare i costi associati alle vostre risorse? ElastiCache](#)
- [COST3: Dovresti usare un tipo di istanza che supporti il tiering dei dati? Quali sono i vantaggi di una suddivisione dei dati in più livelli? Quando non utilizzare le istanze di suddivisione dei dati in più livelli?](#)

**COST1: Come identificate e tenete traccia dei costi associati alle vostre 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. Assicurarsi di disporre di processi e strumenti per tenere traccia delle risorse create dallo sviluppo alla produzione e al pensionamento aiuta a gestire i costi associati ElastiCache.

Vantaggio a livello di domanda: il monitoraggio continuo di tutti i costi associati al carico di lavoro richiede una profonda comprensione dell'architettura che include ElastiCache come uno dei suoi componenti. Inoltre, è necessario disporre di un piano di gestione dei costi per raccogliere e confrontare l'utilizzo rispetto al budget.

- [Obbligatorio] Istituite un Cloud Center of Excellence (CCoE) con una delle sue carte fondamentali per definire, tracciare e intervenire sulle metriche relative all'utilizzo delle vostre organizzazioni. ElastiCache Se CCoE esiste e funziona, assicurati che sappia leggere e tenere traccia dei costi associati a. ElastiCache Quando vengono create le risorse, utilizzate IAM ruoli e policy per verificare che solo team e gruppi specifici possano creare istanze di 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. CCoEdovrebbe identificare, definire e pubblicare metriche di costo aggiornate regolarmente, mensilmente, sull' ElastiCache utilizzo delle chiavi in base a dati categoriali 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. CCoEè costituito da un team interfunzionale con rappresentanti non esclusivi dei team di ingegneria del software, gestione dei dati, team di prodotto, finanza e leadership dell'organizzazione.

[Risorse]:

- [Create a Cloud Center of Excellence](#)
- [ElastiCachePrezzi Amazon](#)
- [Obbligatorio] Utilizza i tag di allocazione dei costi per tenere traccia dei costi con un basso livello di granularità. Usa AWS Cost Management per visualizzare, comprendere e gestire i AWS costi e l'utilizzo nel tempo.
  1. Utilizza i tag per organizzare le risorse e i tag di allocazione dei costi per tenere traccia AWS dei costi a livello dettagliato. Dopo aver attivato i tag di allocazione dei costi, AWS utilizza i tag di allocazione dei costi per organizzare i costi delle risorse nel report di allocazione dei costi, in modo da semplificare la categorizzazione e il monitoraggio dei costi. AWS AWS fornisce due tipi di tag di allocazione dei costi, un tag AWS generato e un tag definito dall'utente. AWS definisce,



crea e applica AWS automaticamente i tag generati e tu definisci, crea e applica 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. Utilizzate i tag di allocazione dei costi per organizzare la AWS fattura in modo da rispecchiare la vostra struttura dei costi. Quando aggiungi tag di allocazione dei costi alle tue risorse in Amazon ElastiCache, sarai in grado di tenere traccia dei costi raggruppando le spese sulle fatture in base ai valori dei tag di risorsa. Puoi anche combinare i tag per monitorare i costi con un livello di dettagli maggiore.

[Risorse]:

- [Utilizzo dei tag di AWS allocazione dei costi](#)
  - [Monitoraggio dei costi con i tag di allocazione dei costi](#)
  - [AWS Cost Explorer](#)
- [Migliore] Collega i ElastiCache costi a metriche valide per tutta l'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'uso con Metrics CloudWatch](#)
- [Buono] Mantieni la visibilità up-to-date architetturale e operativa su metriche e costi per l'intero carico di lavoro utilizzato. ElastiCache
    1. Comprendi l'intero ecosistema di soluzioni, ElastiCache tende a far parte di un ecosistema completo di AWS servizi nel loro set tecnologico, dai client a API Gateway, Redshift QuickSight e agli 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 dovrebbe includere l'uso delle politiche di elaborazione, networking, storage, ciclo di vita, raccolta di metriche, nonché gli elementi operativi e funzionali dell'applicazione

ElastiCache

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 la visibilità, la responsabilità, la definizione delle priorità e le risorse è fondamentale per disporre di una strategia di gestione dei costi efficace. ElastiCache

## COST2: In che modo utilizzate gli strumenti di monitoraggio continuo per ottimizzare i costi associati alle vostre risorse? ElastiCache

Introduzione a livello di domanda: è necessario puntare a un giusto equilibrio tra i ElastiCache costi e le metriche relative alle prestazioni delle applicazioni. Amazon CloudWatch offre visibilità sulle principali metriche operative che possono aiutarti a valutare se le tue ElastiCache risorse sono utilizzate in modo eccessivo o insufficiente rispetto alle tue esigenze. Dal punto di vista dell'ottimizzazione dei costi, è necessario capire quando si verifica un eccesso di approvvigionamento ed essere in grado di sviluppare meccanismi appropriati per ridimensionare ElastiCache le risorse, mantenendo 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 ElastiCache risorse sovradimensionate per lunghi periodi di tempo.

- [Obbligatorio] CloudWatch Utilizzalo per monitorare ElastiCache i cluster e analizzare come queste metriche si relazionano ai dashboard di AWS Cost Explorer.
  1. ElastiCache fornisce sia metriche a livello di host (ad esempio, CPU l'utilizzo) sia metriche specifiche del software del motore di cache (ad esempio, accessi e mancati accessi alla cache). Questi parametri vengono misurati e pubblicati per ogni nodo di cache in intervalli di 60 secondi.
  2. ElastiCache le metriche delle prestazioni (CPUUtilization, EngineUtilization SwapUsage CurrConnections, e sfratti) possono indicare la necessità di scalare verso l'alto o verso il basso (utilizzare tipi di nodi di cache più grandi/più piccoli) o verso l'interno e verso l'esterno (aggiungere più/meno shard). 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]:

- [CloudWatch Monitoraggio dell'uso con Metrics](#)

- [Quali parametri è opportuno monitorare?](#)
- [ElastiCachePrezzi Amazon](#)
- [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 storage 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 far funzionare ElastiCache i cluster utilizzando nodi su richiesta 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

**COST3: Dovresti usare un tipo di istanza che supporti il tiering dei dati? 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 che sia selezionato il tipo di istanza corretto è necessario un esame periodico del tempo di inattività dell'ElastiCache oggetto.

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 SSD storage disponibile per istanza, le istanze di data tiering 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 (RedisOSS).
  2. Solo tipi di istanza limitati supportano la suddivisione dei dati in più livelli.
  3. È supportata solo la versione ElastiCache (RedisOSS) 6.2 e successive
  4. Gli articoli di grandi dimensioni non vengono sostituiti con. SSD Gli oggetti superiori a 128 MiB vengono conservati in memoria.

[Risorse]:

- [Tiering di dati](#)
- [ElastiCachePrezzi Amazon](#)
- [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 utilizzati meno frequentemente comporta un piccolo costo in termini di prestazioni, in quanto questi vengono sostituiti con quelli localiSSD. 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 ElastiCache i cluster utilizzando istanze di tiering non basate sui dati fino a quando non si avrà una migliore comprensione delle proprie esigenze.
  2. Analizza il modello di utilizzo dei dati dei ElastiCache cluster.
  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](#)
- [ElastiCachePrezzi Amazon](#)

## Procedure di risoluzione dei problemi e procedure consigliate comuni con ElastiCache

I seguenti argomenti forniscono consigli per la risoluzione di errori e problemi che potrebbero verificarsi durante l'utilizzo. ElastiCache Se trovi un problema che non è elencato qui, puoi utilizzare il pulsante di feedback in questa pagina per segnalarlo.

Per ulteriori consigli sulla risoluzione dei problemi e risposte alle domande di supporto più comuni, visita il [AWS Knowledge Center](#)

Argomenti

- [Problemi di connessione](#)
- [Errori del client Valkey o Redis OSS](#)
- [Risoluzione dei problemi di latenza elevata in Serverless ElastiCache](#)
- [Risoluzione dei problemi di throttling in Serverless ElastiCache](#)
- [Problemi di connessione persistenti](#)
- [Argomenti correlati](#)

## Problemi di connessione

Se non riesci a connetterti alla ElastiCache cache, prendi in considerazione una delle seguenti opzioni:

1. **Uso TLS:** se si verifica un blocco della connessione durante il tentativo di connessione all' ElastiCache endpoint, è possibile che non sia TLS in uso nel client. Se utilizzi ElastiCache Serverless, la crittografia in transito è sempre abilitata. Assicurati che il client utilizzi TLS per connettersi alla cache. [Scopri di più sulla connessione a una cache TLS abilitata.](#)
2. **VPC:** ElastiCache le cache sono accessibili solo dall'interno di unVPC. Assicurati che l'EC2istanza da cui accedi alla cache e la ElastiCache cache siano create nella stessaVPC. In alternativa, devi abilitare il [VPCpeering](#) tra il VPC luogo in cui risiede l'EC2istanza e il VPC luogo 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 EC2 istanza. Vedi [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 Valkey o Redis. OSS [Scopri di più su come configurare l'accesso alle porte qui.](#)

Se la connessione continua a essere difficile, consulta [Problemi di connessione persistenti](#) gli altri passaggi.

## Errori del client Valkey o Redis OSS

ElastiCache Serverless è accessibile solo tramite client che supportano il protocollo in modalità cluster Valkey o RedisOSS. È possibile accedere ai cluster progettati autonomamente dai client in entrambe le modalità, a seconda della configurazione del cluster.

Se riscontri errori nel tuo client, considera quanto segue:

1. **Modalità cluster:** se si verificano CROSSLOT errori o errori con il [SELECT](#) comando, è possibile che si stia tentando di accedere a una cache abilitata alla modalità cluster con un OSS client Valkey o Redis che non supporta il protocollo Cluster. ElastiCache Serverless supporta solo client

che supportano il protocollo cluster Valkey o Redis. OSS Se desideri utilizzare Valkey o Redis OSS in «Cluster Mode Disabled» (CMD), devi progettare il tuo cluster.

2. CROSSLOTerrori: 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 OSS client Valkey o 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

### Risoluzione dei problemi di latenza lato client

Se notate una latenza elevata sul lato client ma nessun aumento corrispondente `CloudWatch SuccessfulReadRequestLatency` e `SuccessfulWriteRequestLatency` metriche che misurano la latenza lato server, considerate quanto segue:

- Assicurati che il gruppo di sicurezza consenta l'accesso alle porte 6379 e 6380: ElastiCache Serverless utilizza la porta 6379 per l'endpoint primario e la porta 6380 per l'endpoint del lettore. Alcuni client stabiliscono la connettività a entrambe le porte per ogni nuova connessione, anche se l'applicazione non utilizza la funzionalità Read from Replica. Se il gruppo di sicurezza non consente l'accesso in entrata a entrambe le porte, la creazione della connessione può richiedere più tempo. Scopri di più su come configurare l'accesso alle porte [qui](#).

### Risoluzione dei problemi di latenza lato server

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 OSS client Valkey o Redis per scalare le letture e ottenere una latenza inferiore. Se abilitata, ElastiCache Serverless tenta di indirizzare le richieste di lettura ai 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 nella AZs stessa cache:** potresti osservare una maggiore latenza lato client se l'applicazione non è distribuita nella AZs stessa cache. Quando crei una cache serverless, puoi fornire le sottoreti da cui l'applicazione accederà alla cache e ElastiCache Serverless crea endpoint in tali sottoreti. VPC Assicurati che l'applicazione sia distribuita nella stessa. AZs 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 TLS connessione abilitata che utilizza il protocollo. TCP 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 Valkey o Redis esistenti. OSS
- **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 Valkey o Redis, inclusi gli script Lua o OSS i comandi su strutture di dati di grandi dimensioni, possono essere eseguiti a lungo. Per identificare questi comandi, pubblica metriche a livello di comando. ElastiCache 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:** ElastiCache con Valkey e RedisOSS, 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 al ECPUs secondo (90.000 al ECPUs 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 throttling in Serverless ElastiCache

Nelle architetture orientate ai servizi e nei sistemi distribuiti, la limitazione della velocità con cui le API chiamate vengono elaborate dai vari componenti del servizio si chiama throttling. 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 Valkey o Redis dispone di nuovi tentativi integrati per le richieste limitate. OSS 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 metrica ElastiCache 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 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:** ElastiCache con Valkey o RedisOSS, una distribuzione non uniforme delle chiavi o delle 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 30.000 ECPUs /secondo, in un carico di lavoro che esegue semplici comandi/. SET GET
- Leggi dalla replica: se l'applicazione lo consente, prendi in considerazione l'utilizzo della funzione «Leggi dalla replica». La maggior parte dei OSS client Valkey o 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 unità al ECPUs secondo su un singolo slot, per carichi di lavoro con semplici comandi/. SET GET

## Problemi di connessione persistenti

I seguenti elementi devono essere verificati durante la risoluzione dei problemi di connettività persistenti con ElastiCache:

### Argomenti

- [Gruppi di sicurezza](#)
- [Rete ACLs](#)
- [Tabelle di instradamento](#)
- [DNSrisoluzione](#)
- [Identificazione dei problemi con la diagnostica lato server](#)
- [Convalida della connettività di rete](#)
- [Limiti relativi alla rete](#)
- [CPUUtilizzo](#)
- [Connessioni terminate dal lato server](#)
- [Risoluzione dei problemi lato client per le 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 (EC2istanza, AWS Lambda funzione, ECS contenitore Amazon, ecc.) e ElastiCache la cache. 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 il client o il ElastiCache lato client hanno raggiunto i limiti.

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 Valkey o Redis. OSS 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 accedere a un ElastiCache cluster in Amazon VPC](#).

## Rete ACLs

Le liste di controllo degli accessi alla rete (ACLs) sono regole senza stato. Il traffico deve essere consentito in entrambe le direzioni (in entrata e in uscita) per avere esito positivo. ACLsLe reti sono assegnate a sottoreti, non a risorse specifiche. È possibile ACL assegnare le stesse risorse a ElastiCache e al client, specialmente se si trovano nella stessa sottorete.

Per impostazione predefinita, la rete ACLs consente tutto il traffico. Tuttavia, è possibile personalizzarle per negare o consentire il traffico. Inoltre, la valutazione delle ACL regole è sequenziale, il che significa che la regola con il numero più basso corrispondente al traffico lo consentirà o lo negherà. La configurazione minima per consentire il traffico Valkey o Redis OSS è:

### Rete lato client: ACL

- Regole in entrata:
- Numero della regola: preferibilmente inferiore a qualsiasi regola di diniego;
- Tipo: TCP regola 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: TCP regola personalizzata;
- Protocollo: TCP
- Intervallo porte: 6379
- Fonte: 0.0.0.0/0 (o le sottoreti del cluster. ElastiCache Tieni presente che l'utilizzo di specifiche IPs può creare problemi in caso di failover o scalabilità del cluster)
- Consenti/Nega: Consenti

#### ElastiCache Rete: ACL

- Regole in entrata:
- Numero della regola: preferibilmente inferiore a qualsiasi regola di diniego;
- Tipo: TCP regola 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: TCP regola 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 specifiche IPs può creare problemi in caso di failover o scalabilità del cluster)
- Consenti/Nega: Consenti

[Per ulteriori informazioni, consulta Rete. ACLs](#)

## Tabelle di instradamento

Analogamente alla reteACLs, ogni sottorete può avere tabelle di routing diverse. Se i client e il ElastiCache cluster si trovano in sottoreti diverse, assicurati che le relative tabelle di routing consentano loro di raggiungersi.

Gli ambienti più complessi, che coinvolgono più VPCs routing dinamici o firewall di rete, possono diventare difficili da risolvere. Consultare [Convalida della connettività di rete](#) per verificare che le impostazioni di rete siano appropriate.

## DNSrisoluzione

ElastiCache fornisce gli endpoint del servizio in base ai DNS nomi. 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.

DNSLe impostazioni personalizzate (ad esempio, il mancato utilizzo del VPC DNS servizio) potrebbero non essere a conoscenza dei nomi ElastiCache fornitiDNS. Assicuratevi che il sistema sia in grado di risolvere correttamente gli ElastiCache endpoint utilizzando strumenti di sistema come dig (come illustrato 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
```

Puoi anche forzare la risoluzione dei nomi tramite il VPC DNS servizio:

```
$ 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:

- **CPUUtilizzo:** Valkey e Redis OSS sono applicazioni multithread. Tuttavia, l'esecuzione di ogni comando avviene in un singolo thread (principale). Per questo motivo, ElastiCache fornisce le metriche `EngineCPUUtilization` e `EngineCPUUtilization`. `EngineCPUUtilization` fornisce l'CPUUtilizzo dedicato al OSS processo Valkey o Redis e `EngineCPUUtilization` l'utilizzo trasversale. vCPUs I nodi con più di una v di CPU solito hanno valori diversi per `EngineCPUUtilization` e `EngineCPUUtilization` il secondo è generalmente più alto. L'elevato livello `EngineCPUUtilization` può essere causato da un numero elevato di richieste o da operazioni complesse che richiedono molto CPU tempo per essere completate. 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 sprecando 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 scoprire automaticamente la topologia del cluster, i nodi di replica, abilitare le operazioni di lettura tramite il comando `READONLY` Valkey o OSS Redis 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 implicano un sovraccarico significativo. CPU Inoltre, l'handshake TCP a tre vie necessario per creare nuove connessioni influirà negativamente sui tempi di risposta complessivi.

Un ElastiCache nodo con migliaia di connessioni `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. Valkey e Redis OSS offrono prestazioni ottimali con un numero limitato di `currConnections`. Il mantenimento `currConnection` dell'ordine di decine o centinaia riduce al minimo l'utilizzo di risorse per supportare connessioni individuali, come i buffer dei client e i CPU cicli di servizio della 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 tasso elevato di richieste di piccole dimensioni tendono a influire maggiormente sull'`CPUUtilizzo` rispetto alla velocità di trasmissione della rete, mentre chiavi più grandi determinano 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 migliore pratica per i carichi di lavoro con uso intensivo di scrittura consiste nell'utilizzare ElastiCache (RedisOSS) con 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 Redis OSS vengono forniti su un singolo thread, il che significa che 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 OSS 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 keyspace.



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, vai ai comandi [Valkey e Redis](#). OSS

Esempi di potenziali problemi:

- **Script Lua:** Valkey e Redis OSS forniscono un interprete Lua integrato, che consente l'esecuzione di script sul lato server. Gli script Lua su Valkey e Redis OSS vengono eseguiti a livello di motore e sono atomici per definizione, il che significa che nessun altro comando o script potrà essere eseguito mentre uno script è in esecuzione. Gli script Lua offrono la possibilità di eseguire più comandi, algoritmi decisionali, analisi dei dati e altro direttamente sul motore. 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 keystore 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. OSS 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 cancellazione sono sincrone e richiederanno molto CPU tempo se l'elenco dei parametri è grande o contiene un elenco, un set, un set ordinato o un hash (strutture di dati che contengono diversi elementi secondari) di grandi dimensioni. 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 a partire da Redis 4. OSS UNLINK deve essere preferito quando possibile. DEL A partire da ElastiCache (RedisOSS) 6x, il `lazy-free-lazy-user-del` parametro fa sì che il DEL comando si comporti come UNLINK quando è abilitato. Per ulteriori informazioni, consulta [Modifiche ai parametri di Redis 6.0 OSS](#).
- **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 OSS 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 influirà sull'CPUUtilizzo.

Sebbene l'CPUUtilizzo da solo non sia la causa dei problemi di connettività, dedicare troppo tempo all'elaborazione di uno o pochi comandi su più chiavi può causare il fallimento di altre richieste e aumentare l'CPUUtilizzo complessivo.

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 OSS anche comandi che agiscono su una o più chiavi, indipendentemente dal tipo di dati. ElastiCache (RedisOSS) 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 di big data: oltre agli hash, altre strutture di dati possono essere intensive. 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 questi comandi, consulta Comandi [Valkey](#) e Redis. OSS

## Convalida della connettività di rete

Dopo aver esaminato le configurazioni di rete relative alla DNS risoluzione, ai gruppi di sicurezza, alla rete ACLs e alle tabelle di routing, la connettività può essere convalidata con Reachability VPC 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'ENIID (Elastic Network Interface Identification) di uno dei ElastiCache nodi disponibili nel tuo VPC. È possibile trovarlo eseguendo le seguenti operazioni:

1. Vai a <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 DNS convalide precedenti.

3. Annota o salva in altro modo l'ENIID. 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. Creare un percorso di analisi a <https://console.aws.amazon.com/vpc/casa?#ReachabilityAnalyzer> e scegliete le seguenti opzioni:
  - Tipo di fonte: scegli l'istanza se il tuo ElastiCache client funziona su un'EC2istanza Amazon 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 (EC2istanza o ENI ID);
  - Tipo di destinazione: scegli Interfaccia di rete e seleziona Elasticache ENI dall'elenco.
  - Porta di destinazione: specifica 6379 per ElastiCache (RedisOSS) o 11211 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 TCP connettività sulla porta di ElastiCache servizio: su Amazon Linux, Nping è disponibile nel pacchetto nmap e può testare la TCP connettività 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 ElastiCache cluster, come illustrato di seguito:

```
$ 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 hanno `nping` esito negativo e i test `VPCReachability Analyzer` vengono superati, chiedi all'amministratore di sistema di esaminare le possibili regole firewall basate su host, le regole di routing asimmetrico 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, conferma se la TLS sessione può essere stabilita con il seguente comando:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

È previsto un risultato esteso se la connessione e la TLS negoziazione hanno esito positivo. Controllare il codice restituito disponibile nell'ultima riga, il valore deve essere `0` (ok). [Se openssl restituisce qualcosa di diverso, controlla il motivo dell'errore in #. https://www.openssl.org/docs/man1.0.2/man1/verify.html](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, controlla 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/e ElastiCache ha la SSL crittografia in transito abilitata;
- L'applicazione supporta TLS/SSL ma non dispone dei flag di configurazione corretti o di autorità di certificazione affidabili;

## 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 `CurrConnections` metriche di 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. Ogniqualevolta 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 OSS 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.
- `NetworkPacketsPerSecondAllowanceExceeded`: 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, è probabile che si verifichino ritardi nella replica, aumento dei byte utilizzati per la cache, riduzione della memoria liberabile, swap e utilizzo elevati. CPU EC2Le istanze Amazon hanno anche limiti di rete che possono essere tracciati tramite i parametri dei [ENADriver](#). Le istanze Linux con supporto di rete avanzato e ENA driver 2.2.10 o versioni successive possono rivedere i contatori dei limiti con il comando:

```
ethtool -S eth0 | grep "allowance_exceeded"
```

## CPUUtilizzo

La metrica di CPU utilizzo è il punto di partenza dell'indagine e i seguenti elementi possono aiutare a restringere i possibili problemi ElastiCache secondari:

- Redis OSS 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 sopra riportato si è verificato il 26 dicembre alle UTC 19:26:07, ha impiegato 4,8 secondi (4,823 ms) per essere completato ed è stato causato dal comando richiesto dal client 1.2.3.4. KEYS

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 PowerShell su Windows con:

```
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 vengono segnalati frequentemente comandi con complessità temporale O (1), controlla gli altri fattori per un CPU utilizzo elevato menzionati in precedenza.

- **Metriche di latenza:** ElastiCache (RedisOSS) 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`, `ttdl`, `type`, `del`, `expire`, `expireat`, `move`, `persist`, `pexpire`
- `ListBasedCmdsLatency`: `lindex`, `len`, `orange`, `blpop`, `brpop`, `brpoplpush`, `linsert`, `lpop`, `push`, `lpushx`, `lrem`, `lset`, `ltrim`, `rpop`, `rpoplpush`, `rpush`, `rpushx`;
- `PubSubBasedCmdsLatency`: `punsubscribe`, `publish`, `pubsub`, annulla sottoscrizione, sottoscrizione, annullamento dell'iscrizione;
- `SetBasedCmdsLatency`: `scard`, `sdiff`, `sinter`, `sismember`, `smembers`, `srandmember`, `union`, `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 di runtime Redis: OSS
  - `info commandstats`: fornisce un elenco di comandi eseguiti dall'avvio del OSS motore Redis, il numero cumulativo di esecuzioni, 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:** le versioni ElastiCache (RedisOSS) precedenti alla 2.8.22 utilizzano 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 OSS 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 CPU utilizzo, portare a tempi di risposta più elevati e, di conseguenza, a timeout dei client durante l'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

La configurazione predefinita ElastiCache (RedisOSS) 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 entro il periodo indicato, il OSS 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. Le configurazioni predefinite ElastiCache (RedisOSS) non limitano 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 (RedisOSS) 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 le istanze Amazon EC2

Il carico e la reattività sul lato client possono influire anche sulle richieste di. ElastiCache EC2 i limiti delle istanze e del sistema operativo devono essere esaminati attentamente durante la risoluzione dei problemi di connettività intermittente o di timeout. Alcuni punti chiave da osservare:

- CPU:

- EC2CPUutilizzo dell'istanza: assicurati che CPU non sia saturo o vicino 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 utilizzi [EC2istanze espandibili, assicurati che il loro saldo di CPU credito non sia esaurito](#). Queste informazioni sono disponibili sulla metrica. CPUcreditBalance CloudWatch
- Brevi periodi di CPU utilizzo elevato possono causare dei 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, consulta [Amazon EC2 Instance Types](#)
  - 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;
    - `contrack_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 persi a causa di richieste eccessive di metadati dell'istanza, tramite. NTP VPC DNS 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 PPS limite può essere raggiunto quando il traffico di rete è composto 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, ad esempio MGET anziché. GET

## Dissezione del tempo necessario per completare una singola richiesta

- In rete: Tcpcdump 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, tcpdump era in esecuzione ed è stato restituito:

```
$ sudo tcpdump -i any -nn port 6379 -tt
tcpdump: 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 precedente possiamo confermare che l'handshake TCP a tre vie è stato completato in 222 microsecondi (918091 - 917869) e 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. OSSPINGstraceUsando lo stesso comando Redis con 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\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).

È stata impiegata una notevole quantità di tempo, circa 20 ms, per istanziare nc ed eseguire la risoluzione del nome (da 697712 a 717890), dopodiché sono stati necessari 2 ms per creare il TCP socket (da 745659 a 747858) e 0,4 ms (da 747858 a 748330) per inviare e ricevere la risposta alla richiesta.

## Argomenti correlati

- [the section called “Best practice e strategie di caching”](#)

# 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 ElastiCache](#)



# Protezione dei dati in Amazon ElastiCache

Il modello di [responsabilità AWS](#) di si applica alla protezione dei dati in AWS ElastiCache (ElastiCache). Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutto il AWS cloud. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. Questo contenuto include le attività di configurazione e gestione della sicurezza per i AWS servizi che utilizzi. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione [Privacy dei dati FAQ](#).

Ai fini della protezione dei dati, ti consigliamo di proteggere le credenziali AWS dell'account 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.
- TLSUtilizzatelo per comunicare con AWS le risorse.
- Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno AWS dei servizi.
- 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). Ciò include quando lavori con ElastiCache o altri AWS servizi che utilizzano la consoleAPI, AWS CLI, o AWS SDKs. Tutti i dati che inserisci ElastiCache o altri servizi potrebbero essere raccolti per essere inclusi nei registri di diagnostica. Quando fornisci un messaggio URL a un server esterno, non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.

## Argomenti

- [Sicurezza dei dati in Amazon ElastiCache](#)

## Sicurezza dei dati in Amazon ElastiCache

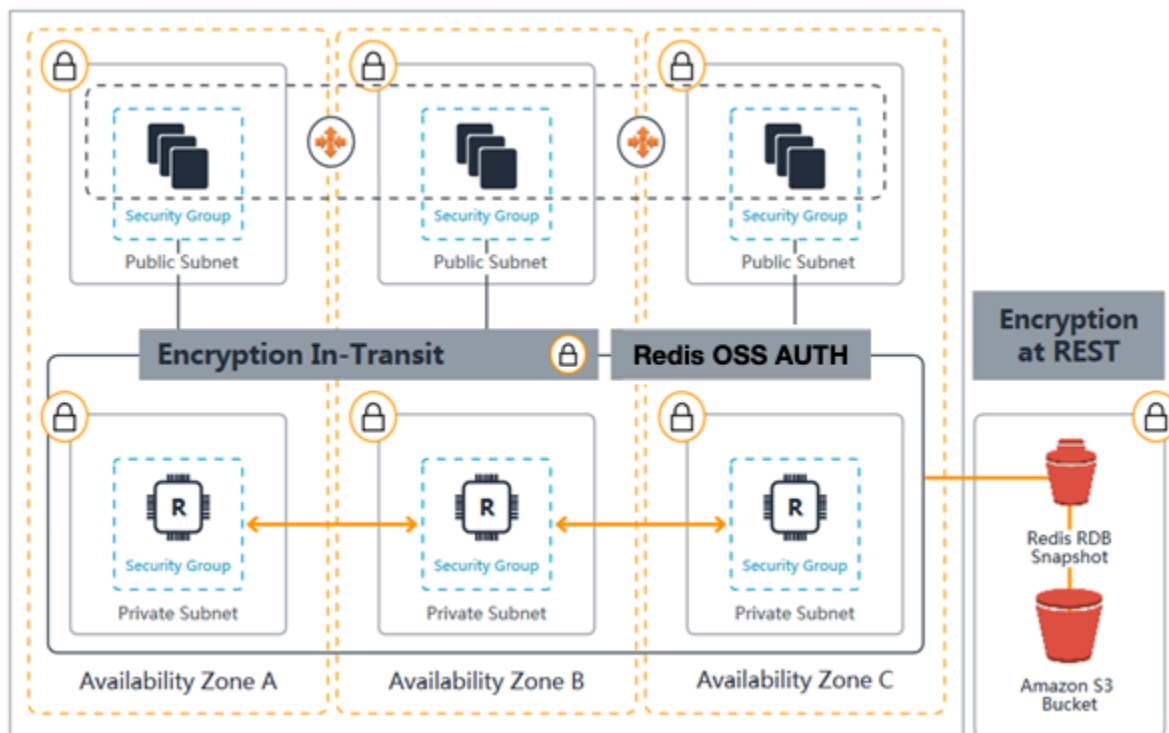
Per aiutarti a proteggere i tuoi dati, Amazon ElastiCache e Amazon EC2 forniscono meccanismi di protezione contro l'accesso non autorizzato ai tuoi dati sul server.

Amazon ElastiCache (Memcached) fornisce funzionalità di crittografia per i dati nelle cache che eseguono le versioni 1.6.12 o successive di Memcached.

Amazon ElastiCache con Valkey e Redis OSS offre funzionalità di crittografia per i dati nelle cache che eseguono Valkey 7.2 o OSS versioni successive e le versioni Redis 3.2.6 (pianificata per, consulta la pianificazione di [fine vita delle OSS versioni Redis](#)) EOL, 4.0.10 o successive. Amazon supporta ElastiCache anche l'autenticazione degli utenti con Valkey e Redis OSS AUTH e l'autorizzazione delle operazioni degli utenti tramite Role-Based Access Control (). IAM RBAC

- 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.

ElastiCache supporta l'autenticazione degli utenti utilizzando i OSS AUTH comandi Valkey e Redis IAM e l'autorizzazione delle operazioni degli utenti tramite Role-Based Access Control (). RBAC



## ElastiCache con Valkey e Redis Security Diagram OSS

### Argomenti

- [ElastiCache crittografia in transito \(\) TLS](#)

- [Crittografia At-Rest in ElastiCache](#)
- [Autenticazione e autorizzazione](#)

## ElastiCache crittografia in transito ( ) TLS

Per aiutarti a proteggere i tuoi dati, Amazon ElastiCache e Amazon EC2 forniscono meccanismi di protezione contro l'accesso non autorizzato ai tuoi dati sul server. Fornendo funzionalità di crittografia in transito, ti ElastiCache offre uno strumento che puoi utilizzare per proteggere i tuoi dati quando vengono spostati da una posizione all'altra.

Tutte le cache OSS serverless Valkey o Redis hanno la crittografia in transito abilitata. Per i cluster progettati autonomamente, è possibile abilitare la crittografia in transito su un gruppo di replica impostando il parametro su `true` (CLI:) quando si crea il gruppo `TransitEncryptionEnabled` di replica. `--transit-encryption-enabled` È possibile eseguire questa operazione indipendentemente dal fatto che si stia creando il gruppo di replica utilizzando il, il o il AWS Management Console. AWS CLI ElastiCache API

Tutte le cache serverless dispongono della crittografia dei dati in transito abilitata. Per i cluster progettati autonomamente, è possibile abilitare la crittografia in transito su un cluster di cache impostando il parametro `TransitEncryptionEnabled` su `true` (CLI: `--transit-encryption-enabled`) quando si crea il cluster di cache utilizzando l'operazione `CreateCacheCluster` (CLI: `create-cache-cluster`

### Argomenti

- [Panoramica della crittografia dei dati in transito](#)
- [Condizioni di crittografia in transito \(Valkey e Redis\) OSS](#)
- [Condizioni di crittografia in transito \(Memcached\)](#)
- [Best practice per la crittografia in transito](#)
- [Altre opzioni Valkey e Redis OSS](#)
- [Abilitazione della crittografia in transito per Memcached](#)
- [Abilitazione della crittografia dei dati in transito](#)
- [Connessione a ElastiCache \(Valkey\) o Amazon ElastiCache \(RedisOSS\) con crittografia in transito tramite valkey-cli](#)
- [Abilitazione della crittografia in transito su un cluster Redis OSS progettato autonomamente utilizzando Python](#)

- [Best practice per abilitare la crittografia dei dati in transito](#)
- [Connessione a nodi abilitati con crittografia in transito tramite Openssl \(Memcached\)](#)
- [Creazione di un client TLS Memcached utilizzando Java](#)
- [Creazione di un client TLS Memcached utilizzando PHP](#)

## Panoramica della crittografia dei dati in transito

La crittografia ElastiCache in transito di Amazon è una funzionalità che consente di aumentare la sicurezza dei 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.

ElastiCache la crittografia in transito implementa le seguenti funzionalità:

- Connessioni client crittografate: le connessioni client ai nodi di cache sono crittografate. 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 le OSS AUTH funzionalità Valkey e Redis, il server può autenticare i client.

## Condizioni di crittografia in transito (Valkey e Redis) OSS

I seguenti vincoli sulla crittografia ElastiCache in transito di Amazon devono essere tenuti presenti quando pianifichi l'implementazione del cluster progettata autonomamente:

- La crittografia in transito è supportata sui gruppi di replica che eseguono Valkey 7.2 e versioni successive e le versioni Redis 3.2.6, 4.0.10 e successive. OSS
- La modifica dell'impostazione di crittografia in transito, per un cluster esistente, è supportata nei gruppi di replica che eseguono Valkey 7.2 e versioni successive e Redis versione 7 e successive. OSS
- La crittografia in transito è supportata solo per i gruppi di replica in esecuzione su Amazon. VPC
- La crittografia in transito non è supportata per i gruppi di replica che eseguono i seguenti tipi di nodi: M1, M2.

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 caching supporti la TLS connettività e di averlo abilitato nella configurazione del client.
- L'utilizzo delle versioni precedenti TLS 1.0 e TLS 1.1 è obsoleto in tutte le AWS regioni per la ElastiCache versione 6 e successive. ElastiCache continuerà a supportare TLS 1.0 e 1.1 fino all'8 maggio 2025. I clienti devono aggiornare il software client prima di tale data.

### Condizioni di crittografia in transito (Memcached)

I seguenti vincoli sulla crittografia ElastiCache in transito di Amazon devono essere tenuti presenti quando pianifichi l'implementazione del cluster progettata autonomamente:

- La crittografia in transito è supportata su cluster che eseguono Memcached versione 1.6.12 e successive.
- La crittografia in transito supporta le versioni 1.2 e 1.3 di Transport Layer Security (TLS).
- La crittografia in transito è supportata solo per i cluster in esecuzione in Amazon. VPC
- La crittografia in transito non è supportata per i gruppi di replica che eseguono i seguenti tipi di nodi: M1, M2, M3, R3, 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`.
- Puoi abilitare la crittografia in transito su un cluster solo durante la creazione del cluster. Non puoi attivare o disattivare la crittografia in transito modificando un cluster.
- Assicurati che il tuo client di caching supporti la TLS connettività e di averlo abilitato 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.

- Poiché la creazione di nuove connessioni può essere costosa, è possibile ridurre l'impatto sulle prestazioni della crittografia in transito mantenendo le connessioni persistenti. TLS

## Altre opzioni Valkey e Redis OSS

Per ulteriori informazioni sulle opzioni disponibili per Valkey e RedisOSS, consulta i seguenti link.

- [Crittografia At-Rest in ElastiCache](#)
- [Autenticazione con i comandi Valkey e Redis OSS AUTH](#)
- [Controllo degli accessi basato sui ruoli \(\) RBAC](#)
- [Amazon VPCs e la ElastiCache sicurezza](#)
- [Identity and Access Management per Amazon ElastiCache](#)

## Abilitazione della crittografia in transito per Memcached

Per abilitare la crittografia in transito creando un cluster Memcached tramite AWS Management Console, effettua le selezioni seguenti:

- Scegli Memcached come motore.
- Scegli la versione del motore 1.6.12 o successiva.
- In Encryption in transit (Crittografia in transito), scegli Enable (Abilita).

Per il step-by-step processo, vedi. [Creazione di un cluster per Valkey o Redis OSS](#)

## 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 in transito utilizzando il AWS Management Console, il, o il AWS CLI. ElastiCache API

## Abilitazione della crittografia in transito utilizzando il AWS Management Console

Abilitazione della crittografia in transito per un nuovo cluster progettato autonomamente utilizzando 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 step-by-step processo, consulta quanto segue:

- [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)
- [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)

Attivazione della crittografia in transito per un cluster esistente progettato autonomamente utilizzando 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 OSS client Valkey o Redis di connettersi utilizzando connessioni crittografate e non crittografate. Dopo aver migrato tutti i OSS client Valkey o Redis per utilizzare connessioni crittografate, è possibile modificare la configurazione del cluster per impostare la modalità di crittografia di 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.

Imposta la modalità di crittografia Transit su Preferred

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Scegli le cache Valkey o le OSS cache Redis dalle ElastiCache Risorse elencate nel pannello di navigazione, presente a sinistra.
3. Scegli la cache che desideri 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 migrato tutti i OSS client Valkey o Redis per utilizzare connessioni crittografate:

## Imposta la modalità di crittografia Transit su Obbligatoria

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Scegli le cache Valkey o le OSS cache Redis dalle ElastiCache Risorse elencate nel pannello di navigazione, presente a sinistra.
3. Scegli la cache che desideri 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 in transito utilizzando il AWS CLI

Per abilitare la crittografia in transito durante la creazione di un gruppo di OSS replica Valkey o Redis utilizzando il, utilizzare il AWS CLI parametro. `transit-encryption-enabled`

Abilitazione della crittografia in transito su un nuovo cluster progettato autonomamente per Valkey o Redis OSS (modalità cluster disabilitata) () CLI

Utilizzate l' AWS CLI operazione `create-replication-group` e i seguenti parametri per creare un gruppo di replica Valkey o Redis con OSS repliche con crittografia in transito abilitata:

Parametri chiave:

- **--engine**—Deve essere o. `valkey redis`
- **--engine-version**—Se il motore è RedisOSS, deve essere 3.2.6, 4.0.10 o successivo.
- **--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 Valkey o Redis OSS \(modalità cluster disabilitata\) \(\)AWS CLI](#)
- [create-replication-group](#)



## Abilitazione della crittografia in transito su un nuovo cluster progettato autonomamente per Valkey o Redis (modalità cluster abilitata) () OSS CLI

Utilizzate l' AWS CLI operazione `create-replication-group` e i seguenti parametri per creare un gruppo di replica Valkey o Redis OSS (abilitata in modalità cluster) con crittografia in transito abilitata:

Parametri chiave:

- **--engine**—Deve essere `o. valkey redis`
- **--engine-version**—Se il motore è RedisOSS, deve essere 3.2.6, 4.0.10 o successivo.
- **--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 tutti 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 Valkey o Redis OSS \(Cluster Mode Enabled\) \(\)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 OSS client Valkey o Redis di connettersi utilizzando connessioni crittografate e non crittografate. Dopo aver migrato tutti i OSS client Valkey o Redis per utilizzare connessioni crittografate, è possibile modificare la configurazione del cluster per impostare la modalità di crittografia di 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.

Utilizzate l' AWS CLI operazione `modify-replication-group` e i seguenti parametri per aggiornare un gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) con la crittografia in transito disabilitata.

Per abilitare la crittografia dei dati in transito

1. Impostare `preferred`, `transit-encryption-mode` utilizzando i seguenti parametri
  - **`--transit-encryption-enabled`**: obbligatorio.
  - **`--transit-encryption-mode`** – Deve essere impostato su `preferred`.
2. `transit-encryption-mode` impostare su `required`, utilizzando i seguenti parametri:
  - **`--transit-encryption-enabled`**: obbligatorio.
  - **`--transit-encryption-mode`** – Deve essere impostato su `required`.

Connessione a ElastiCache (Valkey) o Amazon ElastiCache (RedisOSS) con crittografia in transito tramite `valkey-cli`

Per accedere ai dati dalle cache ElastiCache (RedisOSS) abilitate con la crittografia in transito, si utilizzano client che funzionano con Secure Socket Layer (SSL). Puoi anche usare `valkey-cli` con `TLS` su Amazon Linux e Amazon Linux 2. Se il tuo client non lo supporta `TLS`, puoi usare il comando `stunnel` sull'host del tuo client per creare un SSL tunnel verso i nodi Redis. OSS

Connessione crittografata con Linux

Per utilizzare `valkey-cli` per connettersi a un OSS cluster Valkey o Redis abilitato con crittografia in transito su Amazon Linux 2023, Amazon Linux 2 o Amazon Linux, segui questi passaggi.

1. Scarica e compila l'utilità `valkey-cli`. Questa utilità è inclusa nella distribuzione del software Valkey.
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.

```
valkey-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 https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make distclean
make valkey-cli BUILD_TLS=yes
sudo install -m 755 src/valkey-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 https://github.com/valkey-io/valkey/archive/refs/tags/7.2.6.tar.gz
tar xvzf valkey-7.2.6.tar.gz
cd valkey-7.2.6
make valkey-cli CC=clang BUILD_TLS=yes
sudo install -m 755 src/valkey-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 `valkey-cli`, si consiglia di eseguire il comando opzionale `make-test`
4. Per connetterti a un cluster con crittografia e autenticazione abilitate, inserisci questo comando:

```
valkey-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: `valkey-cli`

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

## Connessione crittografata con stunnel

Per utilizzare `valkey-cli` per connetterti a un OSS cluster Redis abilitato con crittografia in transito tramite `stunnel`, segui questi passaggi.

1. Usalo per connetterti SSH al tuo client e installarlo. `stunnel`

```
sudo yum install stunnel
```

2. Esegui il comando seguente per creare e modificare il file `/etc/stunnel/valkey-cli.conf` contemporaneamente per aggiungere un endpoint del cluster ElastiCache (RedisOSS) a uno o più parametri di connessione, utilizzando l'output fornito di seguito come modello.

```
vi /etc/stunnel/valkey-cli.conf
```

```
fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
```

```
[valkey-cli]
 client = yes
 accept = 127.0.0.1:6379
 connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[valkey-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, `valkey-cli` e `valkey-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 primario è impostato sul OSS valore predefinito di 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 [Ephemeral ports](#) nella Amazon VPC User Guide.
- `connect` è impostato sull'endpoint del server RedisOSS. Per ulteriori informazioni, consulta [Ricerca degli endpoint di connessione in ElastiCache](#).

### 3. Avvia stunnel.

```
sudo stunnel /etc/stunnel/valkey-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. Connect al OSS nodo Redis crittografato utilizzando l'endpoint locale del tunnel.

- Se non è stata utilizzata alcuna AUTH password durante la creazione del cluster ElastiCache (RedisOSS), questo esempio utilizza `valkey-cli` per connettersi al server ElastiCache (RedisOSS) utilizzando il percorso completo per `valkey-cli`, su Amazon Linux:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
```

Se è stata utilizzata AUTH la password durante la creazione del OSS cluster Redis, questo esempio utilizza valkey-cli per connettersi al OSS server Redis utilizzando il percorso completo per valkey-cli, su Amazon Linux:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

O

- Cambia la directory in redis-7.2.5 e procedi come segue:

Se non è stata utilizzata alcuna AUTH password durante la creazione del cluster ElastiCache (RedisOSS), questo esempio utilizza valkey-cli per connettersi al server ElastiCache (RedisOSS) utilizzando il percorso completo per valkey-cli, su Amazon Linux:

```
src/valkey-cli -h localhost -p 6379
```

Se è stata utilizzata AUTH la password durante la creazione del OSS cluster Redis, questo esempio utilizza valkey-cli per connettersi al OSS server Valkey o Redis utilizzando il percorso completo per valkey-cli, su Amazon Linux:

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

Questo esempio utilizza Telnet per connettersi al server Valkey Redis. OSS

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Per arrestare e chiudere i SSL tunnel, `kill` il processo `stunnel`.

```
sudo pkill stunnel
```

## Abilitazione della crittografia in transito su un cluster Redis OSS progettato autonomamente utilizzando Python

La seguente guida mostrerà come abilitare la crittografia in transito su un cluster Redis OSS 7.0 originariamente creato con la crittografia in transito disabilitata. TCPe TLS i client 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 seguente è stato elaborato da un'EC2istanza lanciata nello stesso e VPC che verrà utilizzata per creare il OSS cluster ElastiCache Redis al suo interno.

### Note

- Gli esempi seguenti utilizzano boto3 SDK per le operazioni di ElastiCache gestione (creazione di cluster o utenti) e `redis-py-cluster redis-py/` per la gestione dei dati.
- È necessario utilizzare almeno la versione boto3 (=~) 1.26.39 per utilizzare la migrazione online con la modifica del cluster. TLS API
- ElastiCache supporta la TLS migrazione online solo per i cluster con Valkey versione 7.2 e successive o Redis versione 7.0 o successiva. OSS Quindi, se hai un cluster che esegue una OSS versione Redis precedente alla 7.0, dovrai aggiornare la versione Redis del tuo cluster. OSS Per ulteriori informazioni sulle differenze di versione, consultare [Principali differenze di comportamento e compatibilità delle versioni con Redis OSS](#).

### Argomenti

- [Definisci le costanti di stringa che avvieranno Valkey o Redis Cluster ElastiCache OSS](#)
- [Definizione delle classi per la configurazione del cluster](#)
- [Definizione di una classe che rappresenterà il cluster stesso](#)
- [\(Facoltativo\) Crea una classe wrapper per la connessione del client dimostrativo al cluster Valkey o Redis OSS](#)

- [Creazione della funzione principale che illustra il processo di modifica della configurazione della crittografia dei dati in transito](#)

Definisci le costanti di stringa che avvieranno Valkey o Redis Cluster ElastiCache OSS

Per prima cosa, definiamo alcune semplici costanti di stringa Python che conterranno i nomi delle AWS entità necessarie per creare il ElastiCache cluster come `security-groupCache Subnet group`, e `a. default parameter group`. Tutte queste AWS entità devono essere create in anticipo nel tuo AWS account nella regione che intendi 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 rappresenteranno una configurazione di un cluster, che conterranno metadati sul cluster come la OSS versione Valkey o Redis, il tipo di istanza e se la crittografia 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 lo stesso ElastiCache Valkey o Redis OSS Cluster. Questa classe avrà un campo client che conterrà un client boto3 per operazioni di ElastiCache gestione come la creazione del cluster e l'interrogazione di. ElastiCache API

```

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) Crea una classe wrapper per la connessione del client dimostrativo al cluster Valkey o Redis OSS

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. Crea il cluster utilizzando il client boto3. ElastiCache
2. Inizializza il `redis-py-cluster` client che si conatterà al cluster con una connessione chiara TCP senza. TLS
3. Il client `redis-py-cluster` precompila il cluster con alcuni dati.
4. Il client boto3 attiverà la TLS migrazione da no- TLS a preferito. TLS
5. Durante la migrazione del cluster verso `TLSPreferred`, il `redis-py-cluster` TCP client invierà get operazioni ripetute al cluster fino al termine della migrazione.
6. Al termine della migrazione verso `TLSPreferred`, affermeremo che il cluster supporta la crittografia in transito. Successivamente, creeremo un `redis-py-cluster` client con cui connettersi al cluster. TLS
7. Invieremo alcuni get comandi utilizzando il nuovo TLS client e il vecchio TCP client.
8. Il client boto3 attiverà la TLS migrazione da `TLS Preferred` a `TLS required`.

9. Durante la migrazione del cluster a TLS Required, il redis-py-cluster TLS client invierà get operazioni ripetute al cluster fino al termine della migrazione.

```
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 in transito: assicurati di avere una corretta gestione DNS dei record

### Note

Stiamo modificando ed eliminando i vecchi endpoint durante questo processo. L'uso scorretto degli endpoint può far sì che il OSS client Valkey o Redis utilizzi endpoint vecchi ed eliminati che ne impediranno la connessione al cluster.

Durante la migrazione del cluster da no-TLS a TLS-preferred, i vecchi record per nodo vengono conservati e i nuovi DNS record per nodo vengono generati in un formato diverso DNS. TLS cluster abilitati utilizzano un formato di record diverso rispetto ai cluster. DNS non-TLS-enabled ElastiCache conserverà entrambi i DNS record quando un cluster è configurato in modalità di crittografia: preferibile in modo che Applications e altri OSS client Valkey o Redis possano passare da uno all'altro. Le seguenti modifiche nei DNS record avvengono durante il processo di migrazione TLS:

Descrizione delle modifiche nei DNS record che avvengono quando si abilita la crittografia in transito

Per CME i cluster

Quando un cluster è impostato sulla 'modalità di crittografia dei dati in transito: preferred':

- Gli endpoint del cluster originali per i cluster non TLS abilitati rimarranno attivi. Non si verificheranno tempi di inattività quando il cluster verrà riconfigurato dalla modalità di TLS crittografia da «nessuna» a «preferita».



- I nuovi OSS endpoint TLS Valkey o Redis verranno generati quando il cluster è impostato sulla modalità -preferred. TLS Questi nuovi endpoint si risolveranno IPs come quelli precedenti (non-). TLS
- Il nuovo endpoint di OSS configurazione TLS Valkey o Redis verrà esposto nella ElastiCache console e nella risposta a. `describe-replication-group` API

Quando un cluster è impostato sulla 'modalità di crittografia dei dati in transito: required':

- I vecchi endpoint non TLS abilitati verranno eliminati. Non ci saranno tempi di inattività degli endpoint del TLS cluster.
- È possibile recuperarne uno nuovo `cluster-configuration-endpoint` dalla ElastiCache console o dal. `describe-replication-group` API

Per CMD i cluster 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 cluster non TLS abilitati rimarranno attivi.
- Quando il cluster viene impostato sulla modalità, verranno generati nuovi endpoint TLS primari e di lettura. TLS Preferred Questi nuovi endpoint verranno risolti sugli stessi IP di quelli precedenti (non-TLS).
- Il nuovo endpoint primario e l'endpoint di lettura verranno esposti nella ElastiCache Console e nella risposta a. `describe-replication-group` API

Quando il gruppo di replica è impostato sulla 'modalità di crittografia dei dati in transito: required':

- I vecchi endpoint non TLS primari e di lettura verranno eliminati. Non ci saranno tempi di inattività degli endpoint del TLS cluster.
- È possibile recuperare nuovi endpoint primari e di lettura dalla ElastiCache Console o da. `describe-replication-group` API

L'utilizzo suggerito dei record DNS

Per i CME cluster

- Utilizza l'endpoint di configurazione del cluster anziché i DNS record per nodo nel codice dell'applicazione. Non è consigliabile utilizzare direttamente DNS i nomi per nodo perché potrebbero cambiare quando si aggiungono o si rimuovono gli shard.
- 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. Una volta completata la crittografia in transito, interroga l'endpoint di configurazione del cluster con `describe-replication-group` API (come illustrato sopra (in grassetto)) e utilizza la risposta DNS che otterrai da questo momento in poi.

#### Per i CMD cluster con failover automatico abilitato

- Utilizzate l'endpoint primario e l'endpoint di lettura anziché i nomi per nodo nel codice dell'applicazione, poiché i vecchi DNS nomi per nodo vengono eliminati e ne vengono generati di nuovi durante la migrazione del cluster da `no-` a `DNS -preferred`. TLS L'uso diretto di DNS nomi per nodo non è consigliato perché in futuro potresti aggiungere repliche al tuo cluster. Inoltre, quando il failover automatico è abilitato, i ruoli del cluster primario e delle repliche vengono modificati automaticamente dal ElastiCache servizio. Si consiglia di utilizzare l'endpoint primario e l'endpoint di lettura 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 pratica scorretta, in quanto possono essere modificati durante il processo di migrazione. TLS Una volta completata la modifica della migrazione a `TLS -preferred`, interroga l'endpoint primario e l'endpoint di lettura con il comando che riceverai in risposta da questo momento in `describe-replication-group` API poi. DNS In questo modo sarai in grado di tenere traccia delle modifiche negli endpoint in modo dinamico.

#### Per CMD cluster con failover automatico disabilitato

- Utilizza l'endpoint principale e l'endpoint di lettura anziché i DNS nomi 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 ElastiCache servizio quando il failover automatico è abilitato vengono invece eseguite dall'utente. Ciò consente di tenere traccia dei diversi endpoint manualmente. Poiché i vecchi DNS nomi per nodo vengono eliminati e ne vengono generati di nuovi durante la migrazione del cluster da `no- TLS` a `TLS -preferred`, non utilizzare direttamente i nomi per nodo. DNS Questo è obbligatorio per consentire ai client di connettersi al cluster durante la migrazione. TLS Inoltre, trarrete vantaggio dalla distribuzione

uniforme delle letture tra le repliche quando utilizzate l'endpoint reader e tenete traccia dei DNS record quando aggiungete o eliminate le repliche dal cluster.

- Avere l'endpoint di configurazione del cluster codificato nell'applicazione è una pratica scorretta, in quanto 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 quando il cluster termina la migrazione a TLS -preferred è in grado di accettare e servire entrambe le connessioni. TCP TLS Pertanto, non è necessario creare client che tenteranno di stabilire TLS connessioni al cluster fino al completamento della crittografia in transito.

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 SNS servizio per ricevere una notifica quando la crittografia è completata
- L'utilizzo di `describe-events` API that emetterà un evento al termine della crittografia
- Visualizzazione di un messaggio nella ElastiCache console che indica che la crittografia è stata completata

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")
- Affermare che il cluster è `transit_encryption_enabled` impostato su True interrogando il `describe-replication-group` API

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 dovrebbe aprire TLS le connessioni 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 assicurarti che non ci siano

TCP connessioni più chiare al OSS motore Valkey o Redis usando il comando 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
```

Connessione a nodi abilitati con crittografia in transito tramite Openssl (Memcached)

Per accedere ai dati dai nodi ElastiCache (Memcached) abilitati con la crittografia in transito, è necessario utilizzare client che funzionano con Secure Socket Layer (). SSL Su Amazon linux e Amazon Linux 2, puoi utilizzare anche Openssl s\_client.

Per utilizzare Openssl s\_client per la connessione a un cluster Memcached con crittografia in transito abilitata su Amazon Linux 2 o Amazon Linux:

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

Creazione di un client TLS Memcached utilizzando Java

Per creare un client in TLS modalità, effettuate le seguenti operazioni per inizializzare il client con la modalità appropriata: SSLContext

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
public class TLSDemo {
 public static void main(String[] args) throws Exception {
 ConnectionFactoryBuilder connectionFactoryBuilder = new
ConnectionFactoryBuilder();
 // Build SSLContext
 TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
 tmf.init((KeyStore) null);
```

```

 SSLContext sslContext = SSLContext.getInstance("TLS");
 sslContext.init(null, tmf.getTrustManagers(), null);
 // Create the client in TLS mode
 connectionFactoryBuilder.setSSLContext(sslContext);
 MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

 // Store a data item for an hour.
 client.set("theKey", 3600, "This is the data value");
}
}

```

## Creazione di un client TLS Memcached utilizzando PHP

Per creare un client in TLS modalità, procedi come segue per inizializzare il client con la modalità appropriata: SSLContext

```

<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Discovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 * information
 * about Auto Discovery and persistent-id.
 */

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example */
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
mode */
$tls_client = new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */

```

```
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
 echo $tls_client->getLastErrorMessage(), "\n";
 exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
it to your client.
 * Note: These TLS context configurations will be applied to all the servers connected
to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

/* store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
if($tls_client->set('key', 'value', 60)) {
 print "Successfully stored key\n";
} else {
 echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
 exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
 print "Successfully retrieved key\n";
} else {
 echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
 exit(1);
}
```

Per ulteriori informazioni sull'utilizzo del PHP client, vedere [Installazione del client del cluster ElastiCache per PHP](#).

## Crittografia At-Rest in ElastiCache

Per proteggere i dati, Amazon ElastiCache e Amazon S3 offrono diversi modi per limitare l'accesso ai dati nella cache. Per ulteriori informazioni, consulta [Amazon VPCs e la ElastiCache sicurezza e Identity and Access Management per Amazon ElastiCache](#).

ElastiCache la crittografia at-rest è una funzionalità che aumenta la sicurezza dei dati crittografando i 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 archiviati su SSDs (unità a stato solido) in cluster abilitati al tiering dei dati sono sempre crittografati.

ElastiCache [offre la crittografia predefinita \(gestita dal servizio\) a riposo, oltre alla possibilità di utilizzare le proprie AWS KMS chiavi simmetriche gestite dal cliente in AWS Key Management Service \(\)](#). [KMS](#) 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'attivazione della crittografia At-Rest su un OSS cluster Valkey o Redis esistente progettato autonomamente 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 AWS KMS](#)
- [Abilitazione della crittografia dei dati inattivi](#)
- [Vedi anche](#)

## Condizioni di crittografia dei dati inattivi

I seguenti vincoli sulla crittografia a ElastiCache riposo devono essere tenuti presenti quando pianifichi l'implementazione della crittografia a riposo: ElastiCache

- La crittografia a riposo è supportata nei gruppi di replica che eseguono Valkey 7.2 e versioni successive e nelle versioni Redis (prevista la 3.2.6, vedi la pianificazione di fine del ciclo di vita delle OSS versioni [Redis OSS](#))EOL, 4.0.10 o successive.
- La crittografia a riposo è supportata solo per i gruppi di replica in esecuzione in 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 di esso vengono crittografati indipendentemente dal fatto che la crittografia a riposo sia abilitata o meno.
- L'opzione di utilizzare la chiave gestita dal cliente per la crittografia a riposo 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 vengono crittografati con la AWS KMS chiave gestita dal cliente scelta (o la crittografia gestita dal servizio nelle regioni). SSD AWS GovCloud
- Con Memcached, la crittografia a riposo è supportata solo sulle cache serverless.
- Quando si utilizza Memcached, l'opzione di utilizzare la chiave gestita dal cliente per la crittografia a riposo non è disponibile nelle AWS GovCloud regioni (-1 e -1). us-gov-east us-gov-west

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 AWS KMS

ElastiCache supporta AWS KMS chiavi simmetriche gestite dal cliente (KMSchiave) per la crittografia a riposo. Le KMS chiavi gestite dal cliente sono chiavi di crittografia che crei, possiedi e gestisci nel tuo account. AWS Per ulteriori informazioni, consulta le [AWS KMSchiavi nella AWS Key Management Service Developer Guide](#). Le chiavi devono essere create AWS KMS prima di poter essere utilizzate con ElastiCache.

Per informazioni su come creare chiavi AWS KMS root, consulta [Creating Keys](#) nella AWS Key Management Service Developer Guide.

ElastiCache consente l'integrazione con AWS KMS. Per ulteriori informazioni, consulta [Utilizzo di concessioni](#) nella AWS Guida per gli sviluppatori Key Management Service. Non è necessaria alcuna azione da parte del cliente per abilitare ElastiCache l'integrazione di Amazon con AWS KMS.

La chiave `kms:ViaService` condizionale limita l'uso di una AWS KMS chiave (KMSchiave) alle richieste provenienti da AWS servizi specifici. Da usare `kms:ViaService` con ElastiCache, includi entrambi `ViaService` i nomi nel valore della chiave di condizione: `elasticache.AWS_region.amazonaws.com anddax.AWS_region.amazonaws.com`. Per ulteriori informazioni, vedere [kms: ViaService](#).

Puoi utilizzarlo [AWS CloudTrail](#) per tenere traccia delle richieste a cui Amazon ElastiCache invia per tuo AWS Key Management Service conto. Tutte le API chiamate AWS Key Management Service relative alle chiavi gestite dal cliente hanno CloudTrail i registri corrispondenti. È inoltre possibile visualizzare le sovvenzioni generate ElastiCache chiamando la [ListGrantsKMSAPI](#) chiamata.

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 sono crittografati per impostazione predefinita per utilizzare la KMS chiave 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 il file di backup su un nuovo oggetto S3 e cifrarlo utilizzando una KMS chiave gestita dal cliente, copiare il file in un altro bucket S3 configurato con la crittografia predefinita utilizzando una KMS chiave o modificare un'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 KMS chiave, anche se i dati non sono crittografati nel gruppo di replica originale.

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 ripristinato dopo un guasto hardware. AWS KMS elimina le chiavi principali 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 preserva le proprietà delle AWS KMS chiavi principali, quindi la rotazione non ha alcun effetto sulla capacità di accedere ai dati ElastiCache. ElastiCache Le cache Amazon crittografate non supportano la rotazione manuale delle chiavi, che comporta la creazione di una nuova chiave principale e l'aggiornamento di eventuali riferimenti alla vecchia chiave. Per ulteriori informazioni, consulta [Rotating AWS KMS keys nella AWS Key Management Service Developer Guide](#).
- La crittografia di una ElastiCache cache tramite KMS chiave richiede una concessione per 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 AWS KMS concessioni e limiti, consulta Limits nella AWS Key Management Service Developer Guide.](#)

## 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.

È possibile abilitare la crittografia a riposo quando si crea una ElastiCache cache. È possibile farlo utilizzando il AWS Management Console, il AWS CLI, o il ElastiCache API.

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 della chiave ARN da utilizzare AWS KMS per la crittografia a riposo.

Per informazioni su come creare chiavi AWS KMS root, consulta [Create Keys nella AWS Key Management Service Developer Guide](#)

## Indice

- [Abilitazione della crittografia At-Rest utilizzando il AWS Management Console](#)
- [Abilitazione della crittografia At-Rest utilizzando il AWS CLI](#)

## Abilitazione della crittografia At-Rest su un cluster Valkey o Redis esistente progettato autonomamente OSS

È possibile abilitare la crittografia at-rest solo quando si crea un gruppo di replica Valkey o Redis. OSS 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 in ElastiCache](#) o [Eliminazione di un gruppo di replica](#).

## Abilitazione della crittografia At-Rest utilizzando il 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. Per impostazione predefinita, viene utilizzata una KMS chiave AWS di proprietà per crittografare i dati. Per scegliere la tua AWS KMS chiave, effettua le seguenti selezioni:

- Espandi la sezione Impostazioni predefinite.
- Scegli Personalizza le impostazioni predefinite nella sezione Impostazioni predefinite.
- Scegli Personalizza le impostazioni di sicurezza nella sezione Sicurezza.
- Scegli Customer managed CMK in Encryption key setting.
- 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 gestita dal cliente. CMK

Per la step-by-step procedura, consulta quanto segue:

- [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)
- [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)

Abilitazione della crittografia At-Rest utilizzando il AWS CLI

Per abilitare la crittografia a riposo durante la creazione di un OSS cluster Valkey o Redis utilizzando il AWS CLI, utilizzate il parametro `-- at-rest-encryption-enabled` durante la creazione di un gruppo di replica.

Abilitazione della crittografia At-Rest su un cluster Valkey o Redis OSS (modalità cluster disabilitata) ( ) CLI

L'operazione seguente crea il gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) `my-classic-rg` con tre nodi (`-- num-cache-clusters`), una replica principale e due repliche di lettura. La crittografia a riposo 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 `o. valkey redis`
- `--engine-version`—Se il motore è RedisOSS, deve essere 3.2.6, 4.0.10 o successivo.
- `--at-rest-encryption-enabled`- Richiesto per abilitare la crittografia inattiva.

Example 1: Cluster Valkey o Redis OSS (modalità cluster disabilitata) 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 Valkey o Redis OSS \(modalità cluster disabilitata\) \(AWS CLI\)](#)
- [create-replication-group](#)

Abilitazione della crittografia At-Rest su un cluster per Valkey o Redis OSS (modalità cluster abilitata)  
( ) CLI

L'operazione seguente crea il gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) *my-clustered-rg* con tre gruppi di nodi o shard (--). num-node-groups Ciascuno ha tre nodi, una replica principale e due repliche di lettura (--). replicas-per-node-group La crittografia a riposo 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 o. valkey redis
- **--engine-version**—Se il motore è RedisOSS, deve essere 4.0.10 o successivo.
- **--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 Valkey o Redis OSS (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 Valkey o Redis OSS \(Cluster Mode Enabled\) \(AWS CLI\)](#)
- [create-replication-group](#)

Vedi anche

- [Amazon VPCs e la ElastiCache sicurezza](#)
- [Identity and Access Management per Amazon ElastiCache](#)

## Autenticazione e autorizzazione

AWS Identity and Access Management (IAM) è un servizio Web che consente di controllare in modo sicuro l'accesso alle AWS risorse. ElastiCache supporta l'autenticazione degli utenti utilizzando i OSS AUTH comandi Valkey IAM e Redis e l'autorizzazione delle operazioni degli utenti tramite Role-Based Access Control (). RBAC

### Argomenti

- [Controllo degli accessi basato sui ruoli \(\) RBAC](#)
- [Autenticazione con i comandi Valkey e Redis OSS AUTH](#)
- [Disabilitazione del controllo degli accessi su una cache ElastiCache Valkey o Redis OSS](#)

### Controllo degli accessi basato sui ruoli () RBAC

Invece di autenticare gli utenti con i OSS AUTH comandi Valkey e Redis come descritto in [Autenticazione con i comandi Valkey e Redis OSS AUTH](#), in Valkey 7.2 e versioni successive e Redis OSS 6.0 in poi puoi utilizzare una funzionalità chiamata Role-Based Access Control (). RBAC RBAC è anche l'unico modo per controllare l'accesso alle cache serverless.

A differenza di Valkey e Redis OSSAUTH, in cui tutti i client autenticati hanno accesso completo alla cache se il token è autenticato, RBAC consente di controllare l'accesso alla cache tramite gruppi di utenti. Questi gruppi di utenti sono progettati come un modo per organizzare l'accesso alle cache.

ConRBAC, crei utenti e assegni loro autorizzazioni specifiche utilizzando una stringa di accesso, come descritto di seguito. Gli utenti vengono assegnati a gruppi di utenti allineati a un ruolo specifico (amministratori, risorse umane) che vengono quindi distribuiti in una o più cache. ElastiCache In questo modo, è possibile stabilire limiti di sicurezza tra i client che utilizzano la stessa cache o OSS cache Valkey o Redis e impedire ai client di accedere ai dati degli altri.

#### Note

Quando si utilizza RBAC con i cluster Valkey, sarà comunque necessario assegnare a utenti e gruppi di utenti il motore «redis».

RBAC è progettato per supportare l'introduzione di in Redis 6. [ACLOSS](#) Quando si utilizza RBAC con la OSS cache ElastiCache Valkey o Redis, ci sono alcune limitazioni:



- Non è possibile specificare password in una stringa di accesso. Le password vengono impostate con [CreateUser](#) chiamate. [ModifyUser](#)
- 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 `on` 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 desiderate utilizzare le liste di controllo degli accessi (ACLs) per un comando rinominato, specificate il nome originale del comando, in altre parole il nome del comando prima che fosse rinominato.
- Non è possibile utilizzare `reset` come parte di una stringa di accesso. Specificate le password con API parametri e ElastiCache (RedisOSS) gestisce le password. Pertanto, non è possibile utilizzare `reset` perché rimuoverebbe tutte le password per un utente.
- Redis OSS 6 introduce il comando. [ACLLIST](#) Questo comando restituisce un elenco di utenti insieme alle ACL regole applicate a ciascun utente. ElastiCache (RedisOSS) supporta il ACL LIST comando, ma non include il supporto per gli hash delle password come fa RedisOSS. Con ElastiCache (RedisOSS), puoi utilizzare l'operazione [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 Valkey e Redis ElastiCache includono, e. OSS [ACLWHOAMIACLUSERSACLCAT](#) ElastiCache con Valkey e Redis OSS non supporta nessun altro comando basato sulla scrittura. ACL

- Vengono applicati i vincoli seguenti:

Risorsa	Massimo consentito
Utenti per gruppo di utenti	100
Numero di DPU	1000
Numero di gruppi di utenti	100

L'utilizzo di RBAC with ElastiCache (RedisOSS) è descritto più dettagliatamente di seguito.

## Argomenti

- [Specifica delle autorizzazioni mediante una stringa di accesso](#)

- [Applicazione RBAC a una cache per ElastiCache con Valkey o Redis OSS](#)
- [Migrazione da a AUTH RBAC](#)
- [Migrazione da a RBAC 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 (RedisOSS), create una stringa di accesso e la assegnate a un utente, utilizzando o. AWS CLI 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 informazioni sulla sintassi delle regole, vedere. ACL [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 *comando1*.

+@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 la console e CLI](#).

Se stai migrando un carico di lavoro esistente verso ElastiCache, puoi recuperare la stringa di accesso chiamando `ACL LIST`, escludendo l'utente e qualsiasi hash della password.

Per la OSS versione 6.2 e successive di Redis è supportata anche la seguente sintassi della stringa di accesso:

- `&*` – L'accesso è consentito a tutti i canali disponibili.

Per la OSS versione Redis 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 [le autorizzazioni ACL 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. Vedi i [ACLselettori](#) per ulteriori informazioni.
- `clearselectors` – Elimina tutti i selettori collegati all'utente.

## Applicazione RBAC a una cache per ElastiCache con Valkey o Redis OSS

Per utilizzarlo ElastiCache con Valkey o Redis OSSRBAC, 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 la console e CLI](#)
- [Gestione dei gruppi di utenti con la 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 la console e CLI

Le informazioni utente per RBAC gli utenti 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 predefinito con ID utente e nome `default` utente e lo aggiunge a tutti i gruppi di utenti. Non è possibile eliminare o modificare questo utente. Questo utente è progettato per garantire la compatibilità con il comportamento predefinito delle OSS versioni precedenti di Redis e dispone di una stringa di accesso che gli 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 a AWS Management Console e apri la ElastiCache console Amazon 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 con 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, tieni presente questi vincoli relativi alla password utente quando usi RBAC for ElastiCache (RedisOSS):

- Le password devono essere da 16 a 128 caratteri stampabili.
- I seguenti caratteri non alfanumerici non sono consentiti: , " " / @.

Gestire gli utenti con la console e CLI

Utilizza la procedura seguente per gestire gli utenti sulla console.

Per gestire gli utenti nella console

1. Accedi a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nella ElastiCache dashboard di Amazon, scegli Gestione utenti. 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 il 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 il 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 la 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 a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Nella ElastiCache dashboard di Amazon, scegli Gestione dei 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 seguenti procedure per gestire i gruppi di utenti utilizzando CLI.

Per creare un nuovo gruppo di utenti e aggiungere un utente utilizzando il 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 i membri correnti utilizzando il 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 utilizzando il 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 richiamare l'[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 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 utilizzando il AWS Management Console, procedi come segue:

- Per la modalità cluster disabilitata, vedere [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)
- Per la modalità cluster abilitata, consulta [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)

## Assegnazione di gruppi di utenti alle cache serverless utilizzando il AWS CLI

La seguente AWS CLI operazione crea una cache serverless utilizzando il `user-group-id` parametro con il valore. *my-user-group-id* Sostituisci il gruppo di sottoreti `sng-test` con uno esistente.

### Parametri chiave

- **--engine**— Deve essere `valkey` o `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"
```

La seguente AWS CLI operazione modifica una cache serverless con il `user-group-id` parametro con il 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 ElastiCache degli eventi](#).

### Assegnazione di gruppi di utenti ai gruppi di replica

Dopo aver creato un gruppo di utenti e aggiunto gli utenti, il passaggio finale dell'implementazione 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 utilizzando il AWS Management Console, procedi come segue:

- Per la modalità cluster disabilitata, vedere [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)
- Per la modalità cluster abilitata, consulta [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#)

### Assegnazione di gruppi di utenti a gruppi di replica utilizzando il AWS CLI

La seguente AWS CLI operazione crea un gruppo di replica con la crittografia in transit (TLS) abilitata e il `user-group-ids` parametro con il valore `my-user-group-id`. Sostituisci il gruppo di sottoreti `sng-test` con uno esistente.

### Parametri chiave

- **--engine**— Deve essere `valkey` o `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 AWS CLI operazione modifica un gruppo di replica con la crittografia in transit (TLS) abilitata e il `user-group-ids` parametro 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 ElastiCache degli eventi](#).

## Migrazione da a AUTH RBAC

Se si utilizza AUTH come descritto in [Autenticazione con i comandi Valkey e Redis OSS AUTH](#) e si desidera migrare a UsingRBAC, utilizzare le seguenti procedure.

Utilizzare la procedura seguente per migrare dalla console AUTH all'RBACutilizzo della console.

Per migrare da Valkey o Redis OSS AUTH all'utilizzo della console RBAC

1. Accedi a AWS Management Console e apri la console all' ElastiCache indirizzo. <https://console.aws.amazon.com/elasticache/>
2. Dall'elenco nell'angolo in alto a destra, scegli la AWS regione in cui si trova la cache che desideri 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.

Usa la seguente procedura per migrare da Valkey o Redis all'utilizzo di. OSS AUTH RBAC CLI

## Per migrare da a utilizzare il AUTH RBAC CLI

- Usa il comando `modify-replication-group` come mostrato nell'esempio seguente.

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 a RBAC AUTH

Se utilizzi RBAC e desideri migrare a Redis OSSAUTH, consulta. [Migrazione da a RBAC AUTH](#)

### Note

Se devi disabilitare il controllo degli accessi su una ElastiCache cache, dovrai farlo tramite AWS CLI. Per ulteriori informazioni, consulta [the section called “Disabilitazione del controllo degli accessi su una cache ElastiCache Valkey o Redis OSS”](#).

## Rotazione automatica delle password per utenti

Con AWS Secrets Manager, puoi sostituire le credenziali codificate nel codice (comprese le password) con una chiamata a API Secrets Manager per 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 (RedisOSS) (ovvero segrete) utilizzando una AWS Lambda funzione fornita da Secrets Manager.

[Per ulteriori informazioni su AWS Secrets Manager, consulta What is? AWS Secrets Manager](#)

Come ElastiCache utilizza i segreti

Valkey 7.2 ha un set di funzionalità equivalente a Redis OSS 7.0. In Redis OSS 6, ElastiCache introdotto [Controllo degli accessi basato sui ruoli \(RBAC\)](#) per proteggere il cluster Valkey o Redis. OSS Questa funzionalità consente di limitare determinate connessioni in termini di comandi che è possibile eseguire e tasti a cui è possibile accedere. ConRBAC, mentre il cliente crea un utente con password, i valori della password devono essere inseriti manualmente in testo semplice e sono visibili all'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 [In che modo ElastiCache gli utenti vengono associati al segreto](#).

L'utilizzo di segreti comporta un costo. Per informazioni sui prezzi, consulta [Prezzi di AWS Secrets Manager](#).

In che modo ElastiCache gli utenti vengono associati al segreto

Secrets Manager manterrà un riferimento per l'utente associato nel campo SecretString del segreto. Non ci sarà alcun riferimento al segreto da ElastiCache parte di Side.

```
{
 "password": "strongpassword",
 "username": "user1",
 "user_arn": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918: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, creerai una funzione Lambda che interagirà con il [modify-user](#) API per aggiornare le password dell'utente.

Per informazioni sul funzionamento, consulta [Come funziona la rotazione](#).

**Note**

Per alcuni AWS servizi, per evitare il confuso scenario sostitutivo, si AWS consiglia di utilizzare sia le chiavi di condizione sia le `aws:SourceArn` chiavi di condizione globali. `aws:SourceAccount` Tuttavia, se si include la `aws:SourceArn` condizione nella politica della funzione di rotazione, la funzione di rotazione può essere utilizzata solo per ruotare il segreto da essa ARN specificato. 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, consulta [Risoluzione dei problemi relativi alla rotazione di AWS Secrets Manager](#).

Come creare un ElastiCache utente 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 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 — L'Amazon Resource Name (ARN) del segreto che hai creato nel passaggio 2.
  - iii. USER\_NAME — Nome utente dell' ElastiCache utente,
  - iv. Seleziona Salva.
- g. Seleziona Permissions (Autorizzazioni)
- h. In Ruolo di esecuzione, scegli il nome del ruolo della funzione Lambda da visualizzare sulla IAM console.
- 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:xxxxxxxxxxx918: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:xxxxxxxxxxxx:secret:XXXX"
 },
 {
 "Effect": "Allow",
 "Action": "secretsmanager:GetRandomPassword",
 "Resource": "*"
 }
]
}
```

4. Impostazione della rotazione segreto di Secrets Manager
  - a. Utilizzo di AWS Management Console, vedi [Impostare la rotazione automatica per AWS i segreti di 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 il AWS CLI, vedere [Impostare la rotazione automatica per AWS Secrets Manager l'utilizzo di AWS Command Line Interface](#)

## Autenticazione con IAM

### Argomenti

- [Panoramica](#)

- [Limitazioni](#)
- [Installazione](#)
- [Connessione](#)

## Panoramica

Con IAM Authentication è possibile autenticare una connessione ElastiCache con Valkey o Redis OSS utilizzando AWS IAM identità, quando la cache è configurata per utilizzare Valkey o Redis versione 7 o successiva. OSS Ciò consente di consolidare il modello di sicurezza e semplificare molte attività di sicurezza amministrative. Puoi anche utilizzare l'IAMautenticazione per configurare un controllo granulare degli accessi per ogni singolo ElastiCache utente e cache, seguendo i principi delle autorizzazioni con privilegi minimi. ElastiCache IAMl'autenticazione ElastiCache con Valkey o Redis OSS funziona fornendo un token di IAM autenticazione di breve durata anziché una password utente di lunga durata nel comando Valkey o Redis or. ElastiCache OSS AUTH HELLO Per ulteriori informazioni sul token di IAM autenticazione, consulta il [processo di firma di Signature Version 4](#) nella Guida di riferimento AWS generale e l'esempio di codice riportato di seguito.

È possibile utilizzare IAM le identità e le politiche associate per limitare ulteriormente l'accesso a Valkey o RedisOSS. Puoi anche concedere l'accesso agli utenti dei loro provider di identità federati direttamente alle cache Valkey o Redis. OSS

Per utilizzare AWS IAM con ElastiCache, devi prima creare un ElastiCache utente con la modalità di autenticazione impostata su. IAM È quindi possibile creare o riutilizzare un'IAMidentità. L'IAMidentità necessita di una policy associata per concedere l'elasticache:Connessione alla ElastiCache cache e ElastiCache all'utente. Una volta configurato, è possibile creare un token di IAM autenticazione utilizzando AWS le credenziali dell'IAMutente o del ruolo. Infine, è necessario fornire il token di IAM autenticazione di breve durata come password nel OSS client Valkey o Redis durante la connessione alla cache. Un OSS client Valkey o Redis con supporto per il provider di credenziali può generare automaticamente le credenziali temporanee per ogni nuova connessione. ElastiCache eseguirà IAM l'autenticazione per le richieste di connessione degli ElastiCache utenti IAM abilitati e convaliderà le richieste di connessione con. IAM

## Limitazioni

Quando si utilizza IAM l'autenticazione, si applicano le seguenti limitazioni:

- IAMl'autenticazione è disponibile quando si utilizza ElastiCache con Valkey 7.2 o versione successiva e Redis OSS versione 7.0 o successiva.

- Per ElastiCache gli utenti IAM abilitati, le proprietà nome utente e id utente devono essere identiche.
- Il token di IAM autenticazione è valido per 15 minuti. Per connessioni di lunga durata, consigliamo di utilizzare un OSS client Valkey o Redis che supporti un'interfaccia con un provider di credenziali.
- Una connessione IAM ElastiCache autenticata a Valkey o Redis OSS verrà automaticamente disconnessa dopo 12 ore. La connessione può essere prolungata per 12 ore inviando un HELLO comando AUTH or con un nuovo token di autenticazione. IAM
- IAM l'autenticazione non è supportata nei MULTI EXEC comandi.
- Attualmente, IAM l'autenticazione supporta le seguenti chiavi di contesto delle condizioni globali:
  - Quando si utilizza IAM l'autenticazione con cache serverless, `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime`, e `aws:ResourceTag/%s` (dalle cache e dagli utenti serverless associati) sono supportati.
  - Quando si utilizza IAM l'autenticazione con i gruppi di replica, `aws:SourceIp` e `aws:ResourceTag/%s` (dai gruppi di replica e dagli utenti associati) sono supportati.

Per ulteriori informazioni sulle chiavi di contesto delle condizioni globali, vedere le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'IAM utente.

## Installazione

Per configurare IAM l'autenticazione:

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 sulla politica di IAM fiducia, come mostrato di seguito, per il tuo ruolo che consenta al tuo 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 IAM di policy, come illustrato 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 IAM ruolo.

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

5. Crea la IAM politica.

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

6. Allega la IAM politica 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 IAM abilitato.



```
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 IAM autenticazione di breve durata utilizzando una richiesta [AWS SigV4](#) prefirmata. Dopodiché fornisci il token di IAM autenticazione come password quando ti connetti a una OSS cache Valkey o Redis, come mostrato nell'esempio seguente.

```
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 OSS).
```

```
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);
String iamAuthToken =
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis OSS 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 OSS 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 ElastiCache utilizzando il provider di credenziali di IAM autenticazione.

```
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 OSS).
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
 region, isServerless);

// Create a Redis OSS credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
 RedisIAMAuthCredentialsProvider(
 userId, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis OSS URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
 .withHost(host)
 .withPort(port)
 .withSsl(ssl)
 .withAuthentication(redisCredentialsProvider)
 .build();

// Create a new Lettuce Redis OSS client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Di seguito è riportato un esempio di OSS client Lettuce Redis che inserisce un provider di credenziali per IAMAuthTokenRequest generare automaticamente credenziali temporanee quando necessario.

```
public class RedisIAMAAuthCredentialsProvider implements RedisCredentialsProvider {
 private static final long TOKEN_EXPIRY_SECONDS = 900;

 private final AWSCredentialsProvider awsCredentialsProvider;
 private final String userId;
 private final IAMAuthTokenRequest iamAuthTokenRequest;
 private final Supplier<String> iamAuthTokenSupplier;

 public RedisIAMAAuthCredentialsProvider(String userId,
 IAMAuthTokenRequest iamAuthTokenRequest,
 AWSCredentialsProvider awsCredentialsProvider) {
 this.userName = userName;
 this.awsCredentialsProvider = awsCredentialsProvider;
 this.iamAuthTokenRequest = iamAuthTokenRequest;
 this.iamAuthTokenSupplier =
Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
TimeUnit.SECONDS);
 }

 @Override
 public Mono<RedisCredentials> resolveCredentials() {
 return Mono.just(RedisCredentials.just(userId, iamAuthTokenSupplier.get()));
 }

 private String getIamAuthToken() {
 return
iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
 }
}
```

## Autenticazione con i comandi Valkey e Redis OSS AUTH

### Note

AUTHÈ stato sostituito da [the section called “Controllo degli accessi basato sui ruoli \(\) RBAC”](#) Tutte le cache serverless devono essere utilizzate per l'autenticazione. RBAC

I token o le password di OSS autenticazione Valkey e Redis consentono a Valkey e Redis di richiedere una password prima di consentire OSS ai client di eseguire comandi, migliorando così la sicurezza dei dati. AUTHÈ disponibile solo per i cluster progettati autonomamente.

## Argomenti

- [Panoramica di AUTH in ElastiCache with Valkey e Redis OSS](#)
- [Applicazione dell'autenticazione a un cluster ElastiCache con Valkey o Redis OSS](#)
- [Modifica del AUTH token su un cluster esistente](#)
- [Migrazione da a RBAC AUTH](#)

## Panoramica di AUTH in ElastiCache with Valkey e Redis OSS

Quando lo usi AUTH con il tuo ElastiCache OSS cluster Valkey o Redis, ci sono alcuni perfezionamenti.

In particolare, fai attenzione a questi vincoli relativi a AUTH token o password quando usi: AUTH

- 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 la crittografia in transito abilitata ElastiCache con cluster Valkey o Redis. OSS

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 caratteri:
  - Caratteri maiuscoli
  - Caratteri minuscoli
  - Numeri
  - Caratteri non alfanumerici (!, &, #, \$, ^, <, >, -)
- I token o le password non devono contenere una parola del dizionario o una parola del dizionario leggermente modificata.
- I token o le password non devono essere uguali o simili a quelli di un token usato di recente.

## Applicazione dell'autenticazione a un cluster ElastiCache con Valkey o Redis OSS

È possibile richiedere agli utenti di inserire un token (password) su un server Valkey o Redis protetto da token. OSS A tale scopo, includi il parametro `--auth-token (API:AuthToken)` con il token

corretto quando crei il gruppo o il cluster di replica. Includerlo anche in tutti i comandi successivi per il gruppo di replica o il cluster.

La seguente AWS CLI operazione crea un gruppo di replica con la crittografia in transit (TLS) abilitata e il AUTH token. *This-is-a-sample-token* Sostituisci il gruppo di sottoreti `sng-test` con uno esistente.

#### Parametri chiave

- **--engine**— Deve essere `valkey` o `redis`.
- **--engine-version**— Se il motore è RedisOSS, deve essere 3.2.6, 4.0.10 o successivo.
- **--transit-encryption-enabled**— Richiesto per l'autenticazione e l'idoneità. HIPAA
- **--auth-token**— Richiesto per l'HIPAA idoneità. Questo valore deve essere il token corretto per questo server Valkey o Redis protetto da token. OSS
- **--cache-subnet-group**— Richiesto per l'idoneità. 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 AUTH token su un cluster esistente

Per semplificare l'aggiornamento dell'autenticazione, è possibile modificare il AUTH token utilizzato su un cluster. Puoi apportare questa modifica se la versione del motore è Valkey 7.2 o successiva o Redis 5.0.6 o successiva. ElastiCache deve inoltre avere la crittografia in transito abilitata.

La modifica del token di autenticazione supporta due strategie: ROTATE e SET. La ROTATE strategia aggiunge un AUTH token aggiuntivo al server mantenendo il token precedente. La SET strategia aggiorna il server per supportare un solo AUTH token. Effettuare queste richieste di modifica con il parametro `--apply-immediately` per applicare immediatamente le modifiche.

## Rotazione del token AUTH

Per aggiornare un OSS server Valkey o Redis con un nuovo AUTH token, chiama il `--auth-token` parametro `ModifyReplicationGroup` API with the come nuovo AUTH token e il `--auth-token-update-strategy` con il valore. ROTATE. Una volta completata la ROTATE modifica, il cluster supporterà il AUTH token precedente oltre a quello specificato nel `auth-token` parametro. Se nessun AUTH token era configurato sul gruppo di replica prima della rotazione del AUTH token, il cluster supporta il AUTH token specificato nel `--auth-token` parametro oltre a supportare la connessione senza autenticazione. Vedi [Impostazione del token AUTH](#) per aggiornare il AUTH token da richiedere utilizzando la strategia SET di aggiornamento.

### Note

Se non configuri il AUTH token prima, una volta completata la modifica, il cluster non supporterà alcun AUTH token oltre a quello specificato nel parametro `auth-token`.

Se questa modifica viene eseguita su un server che supporta già due AUTH token, durante questa operazione verrà rimosso anche il AUTH token più vecchio. Ciò consente a un server di supportare fino a due AUTH token più recenti contemporaneamente.

A questo punto, puoi procedere aggiornando il client per utilizzare il AUTH token più recente. Dopo aver aggiornato i client, puoi utilizzare la SET strategia di rotazione dei AUTH token (spiegata nella sezione seguente) per iniziare a utilizzare esclusivamente il nuovo token.

La seguente AWS CLI operazione modifica un 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 OSS server Valkey o Redis in modo che supporti un singolo AUTH token richiesto, chiamate l'ModifyReplicationGroupAPI operazione con il `--auth-token` parametro con lo stesso valore dell'ultimo AUTH token e il `--auth-token-update-strategy` parametro con il valore SET. La SET strategia può essere utilizzata solo con un cluster con 2 AUTH token o 1 AUTH token opzionale utilizzato in ROTATE precedenza. Una volta completata la modifica, il server supporta solo il AUTH token specificato nel parametro `auth-token`.

L' AWS CLI operazione seguente modifica un gruppo di replica su cui impostare il token. AUTH

*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 esistente

Per abilitare l'autenticazione su un OSS server Valkey o Redis esistente, richiama l'ModifyReplicationGroupAPIoperazione. Chiama ModifyReplicationGroup con il --auth-token parametro come nuovo token e il --auth-token-update-strategy con il valore. ROTATE

Una volta completata la ROTATE modifica, il cluster supporta il AUTH token specificato nel --auth-token parametro oltre a supportare la connessione senza autenticazione. Una volta aggiornate tutte le applicazioni client per l'autenticazione su Valkey o Redis OSS con il AUTH token, utilizzate la SET strategia per contrassegnare il AUTH token come richiesto. L'abilitazione dell'autenticazione è supportata solo sui OSS server Valkey e Redis con crittografia in transit () abilitata. TLS

## Migrazione da a RBAC AUTH

Se state autenticando gli utenti con Valkey o Redis OSS Role-Based Access Control (RBAC) come descritto in e desiderate migrare a [Controllo degli accessi basato sui ruoli \(\) RBAC](#), utilizzate le seguenti procedure. AUTH È possibile migrare utilizzando la console o. CLI

Per migrare dalla console RBAC all'AUTHutilizzo della console

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <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. Per Operazioni, scegli Modifica.

Viene visualizzata la finestra Modifica.

6. Per il controllo degli accessi, scegli l'accesso utente AUTH predefinito Valkey o l'accesso utente predefinito Redis OSS AUTH.
7. In token Valkey o AUTH token Redis, imposta un nuovo OSS AUTH token.

8. Scegli Visualizza l'anteprima delle modifiche e quindi, nella schermata successiva, seleziona Modifica.

Per migrare da RBAC a utilizzare il AUTH AWS CLI

Utilizzate uno dei seguenti comandi per configurare un nuovo AUTH token opzionale per il vostro gruppo di replica Valkey o RedisOSS. Tieni presente che un token di autenticazione opzionale consentirà l'accesso non autenticato al gruppo di replica fino a quando il token di autenticazione non sarà contrassegnato come richiesto, utilizzando la strategia di aggiornamento nella fase successiva. SET

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id test \
 --remove-user-groups \
 --auth-token This-is-a-sample-token \
 --auth-token-update-strategy ROTATE \
 --apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id test ^
 --remove-user-groups ^
 --auth-token This-is-a-sample-token ^
 --auth-token-update-strategy ROTATE ^
 --apply-immediately
```

Dopo aver eseguito il comando precedente, è possibile aggiornare le OSS applicazioni Valkey o Redis per l'autenticazione al gruppo di ElastiCache replica utilizzando il token opzionale appena configurato. AUTH Per completare la rotazione del token di autenticazione, utilizza la strategia di aggiornamento SET nel comando successivo di seguito. Questo contrassegnerà il AUTH token opzionale come richiesto. Al termine dell'aggiornamento del token di autenticazione, lo stato del gruppo di replica verrà visualizzato come ACTIVE e tutte le connessioni a questo gruppo di replica richiederanno l'autenticazione.

Per Linux, macOS o Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id test \
 --remove-user-groups \
 --auth-token This-is-a-sample-token \
 --auth-token-update-strategy ROTATE \
 --apply-immediately
```

```
--replication-group-id test \
--auth-token This-is-a-sample-token \
--auth-token-update-strategy SET \
--apply-immediately
```

Per Windows:

```
aws elasticache modify-replication-group ^
--replication-group-id test ^
--remove-user-groups ^
--auth-token This-is-a-sample-token ^
--auth-token-update-strategy SET ^
--apply-immediately
```

Per ulteriori informazioni, consulta [Autenticazione con i comandi Valkey e Redis OSS AUTH](#).

#### Note

Se è necessario disabilitare il controllo degli accessi su un ElastiCache cluster, vedere. [the section called “Disabilitazione del controllo degli accessi su una cache ElastiCache Valkey o Redis OSS”](#)

## Disabilitazione del controllo degli accessi su una cache ElastiCache Valkey o Redis OSS

Segui le istruzioni riportate di seguito per disabilitare il controllo degli accessi su una cache abilitata per Valkey o Redis. OSS TLS La cache avrà uno dei due diversi tipi di configurazioni: accesso utente AUTH predefinito o Elenco di controllo degli accessi al gruppo di utenti (). RBAC Se la cache è stata creata con la AUTH configurazione, è necessario modificarla in base alla RBAC configurazione prima di poter disabilitare la cache rimuovendo i gruppi di utenti. Se la cache è stata creata con la RBAC configurazione, puoi procedere direttamente alla disattivazione.

Per disabilitare una cache OSS serverless Valkey o 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 OSS cache Valkey o Redis con configurato con un token AUTH

1. Cambia il AUTH token 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 AUTH token 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 OSS cluster Valkey o 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>
{
 "..."
```

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 proteggere i dati della cache e proteggerli da accessi non autorizzati:

- [Amazon VPCs e la ElastiCache sicurezza](#) 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.

## Argomenti

- [Amazon VPCs e la ElastiCache sicurezza](#)
- [ElastiCache API e VPC endpoint di interfaccia \(AWS PrivateLink\)](#)
- [Sottoreti e gruppi di sottoreti](#)

## Amazon VPCs e la ElastiCache sicurezza

Poiché la sicurezza dei dati è importante, ti ElastiCache fornisce 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 Amazon Virtual Private Cloud (AmazonVPC) o Amazon EC2 -Classic.

### Important

Abbiamo reso obsoleto l'uso di Amazon EC2 -Classic per l'avvio di cluster. ElastiCache Tutti i nodi della generazione corrente vengono avviati solo in Amazon Virtual Private Cloud.

Il servizio Amazon Virtual Private Cloud (AmazonVPC) definisce una rete virtuale molto simile a un data center tradizionale. Quando configuri Amazon, VPC puoi selezionare l'intervallo di indirizzi IP, creare sottoreti e configurare tabelle di routing, gateway di rete e impostazioni di sicurezza. Puoi anche aggiungere un cluster di cache alla rete virtuale e controllare l'accesso al cluster di cache utilizzando i gruppi VPC di sicurezza Amazon.

Questa sezione spiega come configurare manualmente un ElastiCache cluster in AmazonVPC. Queste informazioni sono destinate agli utenti che desiderano una comprensione più approfondita del modo ElastiCache in cui Amazon VPC collabora.

## Argomenti

- [Comprensione ElastiCache e Amazon VPCs](#)
- [Modelli di accesso per accedere a una ElastiCache cache in Amazon VPC](#)
- [Creazione di un cloud privato virtuale \(VPC\)](#)
- [Connessione a una cache in esecuzione su Amazon VPC](#)



## Comprensione ElastiCache e Amazon VPCs

ElastiCache è completamente integrato con Amazon Virtual Private Cloud (AmazonVPC). Per ElastiCache gli utenti, ciò significa quanto segue:

- Se il tuo AWS account supporta solo la VPC piattaforma EC2 -, avvia ElastiCache sempre il cluster su AmazonVPC.
- Se sei un principiante AWS, i tuoi cluster verranno distribuiti in un Amazon. VPC VPCVerrà creato automaticamente un valore predefinito per te.
- Se hai un'impostazione predefinita VPC e non specifichi una sottorete quando avvii un cluster, il cluster viene avviato nel tuo Amazon predefinito. VPC

Per ulteriori informazioni, consulta [Rilevamento delle piattaforme supportate e se ne hai una predefinita](#). VPC

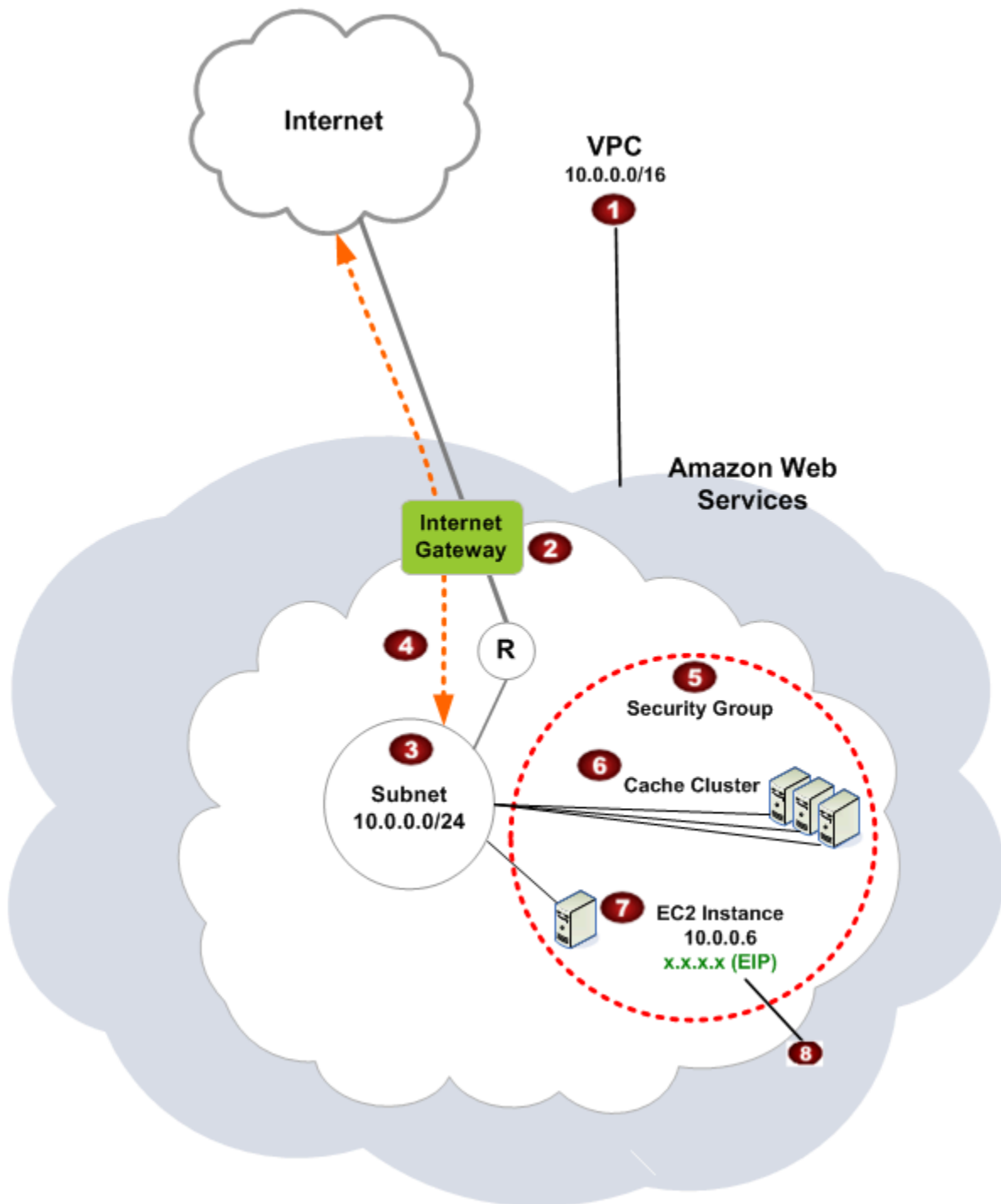
Con Amazon Virtual Private Cloud, puoi creare una rete virtuale nel AWS cloud che assomiglia molto a un data center tradizionale. Puoi configurare AmazonVPC, inclusa la selezione dell'intervallo di indirizzi IP, la creazione di sottoreti e la configurazione di tabelle di routing, gateway di rete e impostazioni di sicurezza.

La funzionalità di base di ElastiCache è la stessa in un cloud privato virtuale; ElastiCache gestisce gli aggiornamenti software, l'applicazione di patch, il rilevamento degli errori e il ripristino indipendentemente dal fatto che i cluster siano distribuiti all'interno o all'esterno di Amazon. VPC

ElastiCache ai nodi di cache distribuiti all'esterno di Amazon VPC viene assegnato un indirizzo IP a cui viene risolto l'endpoint/nomeDNS. Ciò fornisce connettività da istanze Amazon Elastic Compute Cloud (AmazonEC2). Quando avvii un ElastiCache cluster in una sottorete VPC privata Amazon, a ogni nodo di cache viene assegnato un indirizzo IP privato all'interno di quella sottorete.

### Panoramica di ElastiCache in un Amazon VPC

Il diagramma e la tabella seguenti descrivono l'VPCambiente Amazon, insieme ai ElastiCache cluster e alle EC2 istanze Amazon che vengono lanciati in Amazon. VPC



1

Amazon VPC è una porzione isolata del AWS Cloud a cui viene assegnato un proprio blocco di indirizzi IP.

2

Un gateway Internet collega Amazon VPC direttamente a Internet e fornisce l'accesso ad altre AWS risorse come Amazon Simple Storage Service (Amazon S3) che sono in esecuzione all'esterno di Amazon VPC

**3**

Una VPC sottorete Amazon è un segmento dell'intervallo di indirizzi IP di Amazon VPC in cui puoi isolare AWS le risorse in base alle tue esigenze operative e di sicurezza.

**4**

Una tabella di routing in Amazon VPC indirizza il traffico di rete tra la sottorete e Internet. Amazon VPC ha un router implicito, che è simboleggiato in questo diagramma dal cerchio con la R.

**5**

Un gruppo VPC di sicurezza Amazon controlla il traffico in entrata e in uscita per i tuoi ElastiCache cluster e le tue istanze Amazon EC2

**6**

È possibile avviare un ElastiCache cluster nella sottorete. I nodi della cache hanno indirizzi IP privati dall'intervallo di indirizzi della sottorete.

**7**

Puoi anche avviare EC2 istanze Amazon nella sottorete. Ogni EC2 istanza Amazon ha un indirizzo IP privato dall'intervallo di indirizzi della sottorete. L'EC2 istanza Amazon può connettersi a qualsiasi nodo di cache nella stessa sottorete.

**8**

Affinché un'EC2 istanza Amazon nel tuo Amazon VPC sia raggiungibile da Internet, devi assegnare all'istanza un indirizzo pubblico statico chiamato indirizzo IP elastico.

## Prerequisiti

Per creare un ElastiCache cluster all'interno di un AmazonVPC, Amazon VPC deve soddisfare i seguenti requisiti:

- Amazon VPC deve consentire EC2 istanze Amazon non dedicate. Non è possibile ElastiCache utilizzarlo in un Amazon VPC configurato per la tenancy di istanze dedicate.
- È necessario definire un gruppo di sottoreti di cache per AmazonVPC. ElastiCache utilizza quel gruppo di sottorete di cache per selezionare una sottorete e gli indirizzi IP all'interno di quella sottorete da associare agli VPC endpoint o ai nodi di cache.

- CIDRi blocchi per ogni sottorete devono essere sufficientemente grandi da fornire indirizzi IP di riserva da utilizzare durante le attività ElastiCache di manutenzione.

## Routing e sicurezza

Puoi configurare il routing in Amazon VPC per controllare dove scorre il traffico (ad esempio, verso il gateway Internet o il gateway privato virtuale). Con un gateway Internet, Amazon VPC ha accesso diretto ad altre AWS risorse che non sono in esecuzione su AmazonVPC. Se scegli di avere solo un gateway privato virtuale con una connessione alla rete locale della tua organizzazione, puoi instradare il traffico diretto a Internet VPN e utilizzare le politiche di sicurezza locali e il firewall per controllare l'uscita. In tal caso, l'accesso alle risorse tramite Internet comporta costi aggiuntivi per la larghezza di banda. AWS

Puoi utilizzare i gruppi VPC di sicurezza Amazon per proteggere ElastiCache i cluster e EC2 le istanze Amazon nel tuo Amazon. VPC I gruppi di sicurezza operano come un firewall a livello di istanza, non di sottorete.

### Note

Ti consigliamo vivamente di utilizzare DNS i nomi per connetterti ai nodi della cache, poiché l'indirizzo IP sottostante può cambiare.

## VPCDocumentazione Amazon

Amazon VPC dispone di una propria documentazione per descrivere come creare e utilizzare AmazonVPC. La tabella seguente fornisce i collegamenti alle VPC guide di Amazon.

Descrizione	Documentazione
Come iniziare a usare Amazon VPC	<a href="#">Guida introduttiva ad Amazon VPC</a>
Come usare Amazon VPC tramite AWS Management Console	<a href="#">Guida per VPC l'utente di Amazon</a>
Descrizioni complete di tutti i VPC comandi Amazon	<a href="#">Amazon EC2 Command Line Reference</a> (i VPC comandi Amazon si trovano nel EC2 riferimento Amazon)

Descrizione	Documentazione
Descrizioni complete delle VPC API operazioni, dei tipi di dati e degli errori di Amazon	<a href="#">Amazon EC2 API Reference</a> (le VPC API operazioni di Amazon si trovano nel EC2 riferimento Amazon)
Informazioni per l'amministratore di rete che deve configurare il gateway al termine di una IPsec VPN connessione opzionale	<a href="#">Che cos'è AWS Site-to-Site VPN?</a>

Per informazioni più dettagliate su Amazon Virtual Private Cloud, consulta [Amazon Virtual Private Cloud](#).

## Modelli di accesso per accedere a una ElastiCache cache in Amazon VPC

Amazon ElastiCache supporta i seguenti scenari per l'accesso a una cache in AmazonVPC:

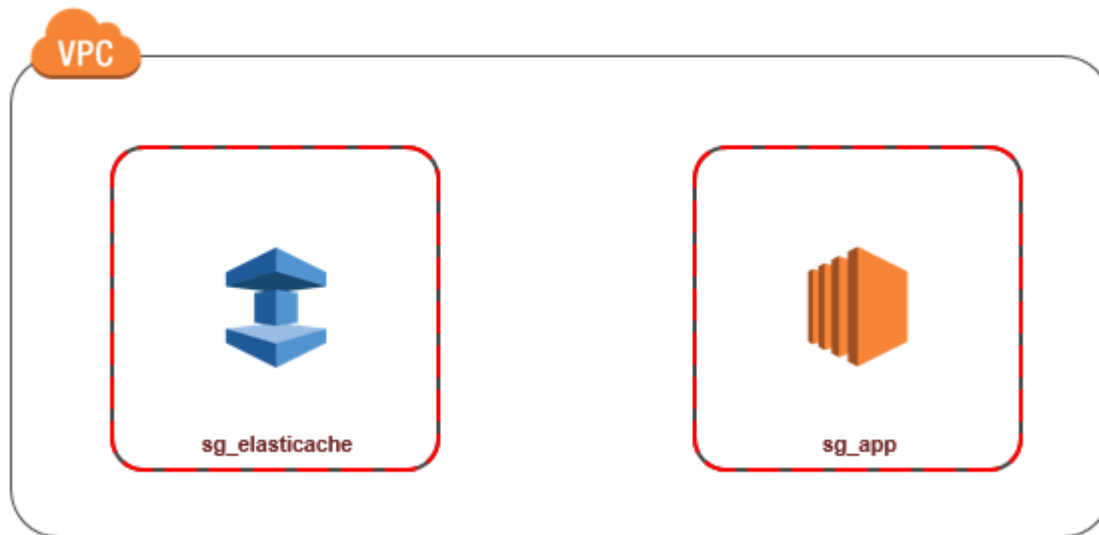
### Indice

- [Accedere a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano nella stessa Amazon VPC](#)
- [Accedere a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in un Amazon diverso VPCs](#)
  - [Accesso a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in Amazon diverse VPCs nella stessa regione](#)
    - [Uso del Transit Gateway](#)
  - [Accedere a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in Amazon diverse VPCs in regioni diverse](#)
    - [Usare Transit VPC](#)
- [Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente](#)
  - [Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente utilizzando VPN la connettività](#)
  - [Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect](#)

Accedere a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano nella stessa Amazon VPC

Il caso d'uso più comune è quando un'applicazione distribuita su un'EC2istanza deve connettersi a una cache della stessaVPC.

Il diagramma seguente illustra questo scenario.



Il modo più semplice per gestire l'accesso tra EC2 istanze e cache all'interno di una stessa istanza VPC consiste nel fare quanto segue:

1. Crea un gruppo VPC di sicurezza per la tua cache. Questo gruppo di sicurezza può essere usato per limitare l'accesso alla cache. Ad esempio, puoi creare una regola personalizzata per questo gruppo di sicurezza che consenta l'accesso TCP utilizzando la porta assegnata alla cache al momento della creazione e un indirizzo IP che utilizzerai per accedere alla cache.

La porta predefinita per le cache Memcached è 11211.

La porta predefinita per le OSS cache Valkey e Redis è 6379.

2. Crea un gruppo VPC di sicurezza per le tue EC2 istanze (server web e applicazioni). Questo gruppo di sicurezza può, se necessario, consentire l'accesso all'EC2 istanza da Internet tramite la tabella VPC di routing dell'istanza. Ad esempio, è possibile impostare regole su questo gruppo di sicurezza per consentire TCP l'accesso all'EC2 istanza tramite la porta 22.
3. Crea regole personalizzate nel gruppo di sicurezza per la tua cache che consentano le connessioni dal gruppo di sicurezza che hai creato per le tue EC2 istanze. In tal modo qualsiasi membro del gruppo di sicurezza può accedere alle cache.

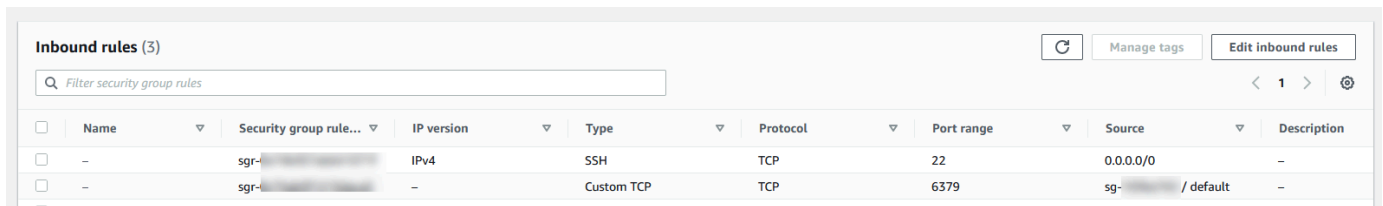
#### Note

Se si prevede di utilizzare [Local Zones](#), assicurati di averlo abilitato. Quando crei un gruppo di sottoreti in quella zona locale, il tuo VPC viene esteso a quella zona locale e tratterai la

sottorete come qualsiasi sottorete in qualsiasi altra zona di disponibilità. VPC Tutti i gateway e le tabelle di routing pertinenti verranno regolati automaticamente.

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

1. Accedi alla Console di AWS gestione e apri la VPC console Amazon all'indirizzo <https://console.aws.amazon.com/vpc>.
2. Fai clic su Security Groups (Gruppi di sicurezza) nel pannello 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. Da Tipo scegli Regola personalizzata TCP.
  - a. Per Intervallo di porte specifica la porta utilizzata alla creazione della cache.  
 La porta predefinita per le cache Memcached è 11211.  
 La porta predefinita per le OSS cache e i gruppi di replica Valkey e Redis è. 6379
  - b. Nella casella Source (fonte) iniziare a digitare l'ID del gruppo di sicurezza. Dall'elenco seleziona il gruppo di sicurezza che utilizzerai per le tue EC2 istanze Amazon.
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	-

Accedere a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in un Amazon diverso VPCs

Quando la cache si trova in un'EC2istanza diversa VPC da quella che stai utilizzando per accedervi, ci sono diversi modi per accedere alla cache. Se la cache e l'EC2istanza si trovano in VPCs aree diverse ma nella stessa regione, puoi utilizzare il VPC peering. Se la cache e l'EC2istanza si trovano in regioni diverse, puoi creare VPN connettività tra le regioni.

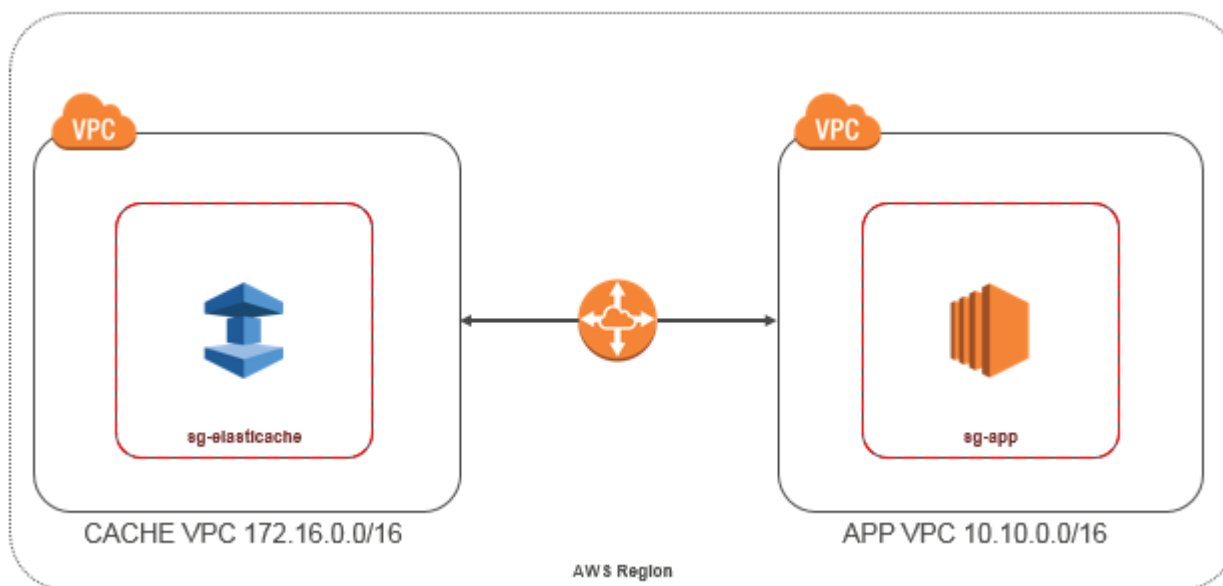
## Argomenti



- [Accesso a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in Amazon diverse VPCs nella stessa regione](#)
- [Accedere a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in Amazon diverse VPCs in regioni diverse](#)

Accesso a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in Amazon diverse VPCs nella stessa regione

Il diagramma seguente illustra l'accesso a una cache da parte di un'EC2istanza Amazon in un'altra Amazon VPC nella stessa regione utilizzando una connessione VPC peering Amazon.



Cache a cui accede un'EC2istanza Amazon in un'altra Amazon VPC all'interno della stessa regione - VPC Peering Connection

Una connessione VPC peering è una connessione di rete tra due VPCs che consente di instradare il traffico tra di esse utilizzando indirizzi IP privati. Le istanze di entrambe VPC possono comunicare tra loro come se si trovassero all'interno della stessa rete. Puoi creare una connessione VPC peering tra il tuo Amazon VPCs o con un Amazon VPC in un altro AWS account all'interno di una singola regione. Per ulteriori informazioni sul VPC peering di Amazon, consulta la [VPCdocumentazione](#).

**Note**

DNSLa risoluzione dei nomi potrebbe non riuscire per il peeredVPCs, a seconda delle configurazioni applicate a. ElastiCache VPC Per risolvere questo problema, entrambi VPCs devono essere abilitati per i DNS nomi host e la risoluzione. DNS Per ulteriori informazioni, consulta [Abilitare la DNS risoluzione per una connessione VPC peering](#).

Per accedere a una cache in un altro Amazon VPC tramite peering

1. Assicurati che i due VPCs non abbiano un intervallo IP sovrapposto o non sarai in grado di peerizzarli.
2. Dai un'occhiata ai due. VPCs Per ulteriori informazioni, consulta [Creazione e accettazione di una connessione Amazon VPC Peering](#).
3. Aggiornare la tabella di routing. Per ulteriori informazioni, consulta [Aggiornamento delle tabelle di routing per una connessione VPC peering](#)

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

VPC Tabella di routing

4. Modifica il gruppo di sicurezza della ElastiCache cache per consentire la connessione in entrata dal gruppo di sicurezza dell'applicazione nel peer. VPC Per ulteriori informazioni, consulta [Reference Peer VPC Security Groups](#).

L'accesso a una cache con una connessione peering implica ulteriori costi di trasferimento dei dati.

Uso del Transit Gateway

Un gateway di transito consente di collegarsi VPCs e di effettuare VPN connessioni nella stessa AWS regione e di instradare il traffico tra di esse. Un gateway di transito funziona su più AWS account

ed è possibile utilizzare AWS Resource Access Manager per condividere il gateway di transito con altri account. Dopo aver condiviso un gateway di transito con un altro AWS account, il proprietario dell'account può collegarlo VPCs al gateway di transito. Un utente di uno qualsiasi degli account può eliminare il collegamento in qualsiasi momento.

È possibile abilitare il multicast su un gateway di transito e quindi creare un dominio multicast del gateway di transito che consenta l'invio del traffico multicast dalla sorgente multicast ai membri del gruppo multicast tramite VPC allegati associati al dominio.

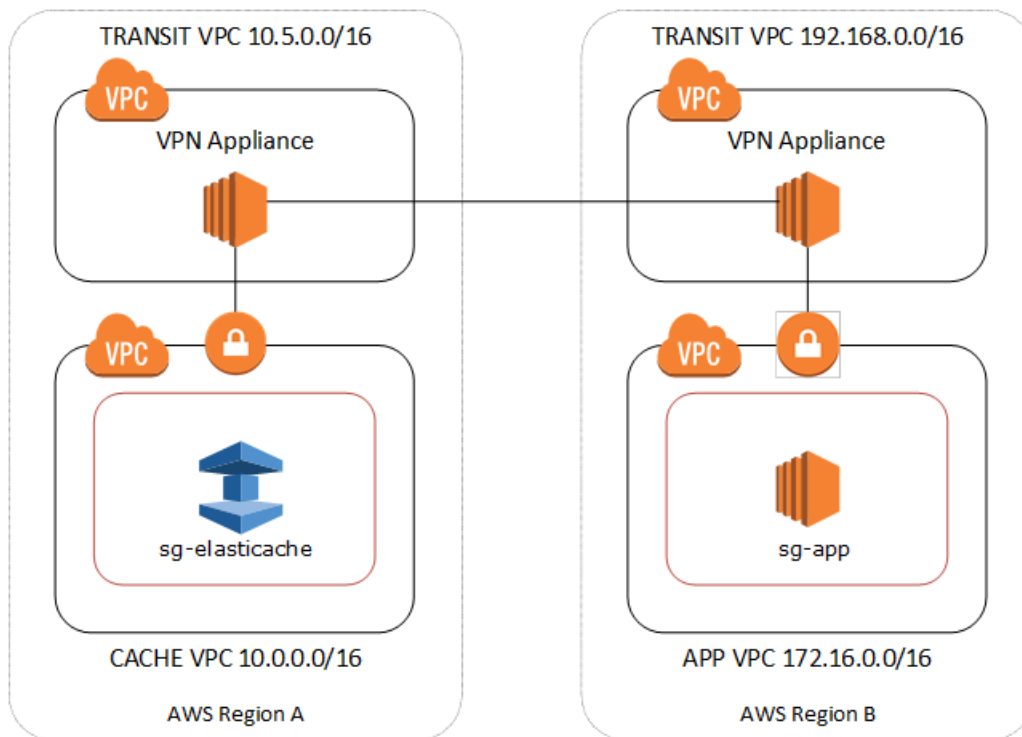
È inoltre possibile creare un allegato di connessione peering tra gateway di transito in diverse regioni. AWS In questo modo è possibile instradare il traffico tra gli allegati dei gateway di transito in diverse regioni.

Per ulteriori informazioni, consulta [Gateway di transito](#).

Accedere a una ElastiCache cache quando essa e l'EC2istanza Amazon si trovano in Amazon diverse VPCs in regioni diverse

Usare Transit VPC

Un'alternativa all'utilizzo del VPC peering, un'altra strategia comune per connettere più reti remote VPCs e geograficamente disperse consiste nella creazione di un transito VPC che funga da centro di transito di rete globale. Un transito VPC semplifica la gestione della rete e riduce al minimo il numero di connessioni necessarie per connettere reti multiple e remote. VPCs 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 diverse VPCs regioni

Una volta creato Transit AmazonVPC, un'applicazione distribuita in un «spoke» VPC in una regione può connettersi a una ElastiCache cache in un «spoke» VPC all'interno di un'altra regione.

Per accedere a una cache in un'altra regione VPC all'interno di un'altra regione AWS

1. Implementa una VPC soluzione di transito. Per ulteriori informazioni, consulta [AWS Transit Gateway](#).
2. Aggiorna le tabelle VPC di routing nell'app e nella cache VPCs per indirizzare il traffico attraverso VGW (Virtual Private Gateway) e l'VPN Appliance. In caso di Dynamic Routing with Border Gateway Protocol (BGP), i percorsi potrebbero essere propagati automaticamente.
3. Modifica il gruppo di sicurezza della ElastiCache cache 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 ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente

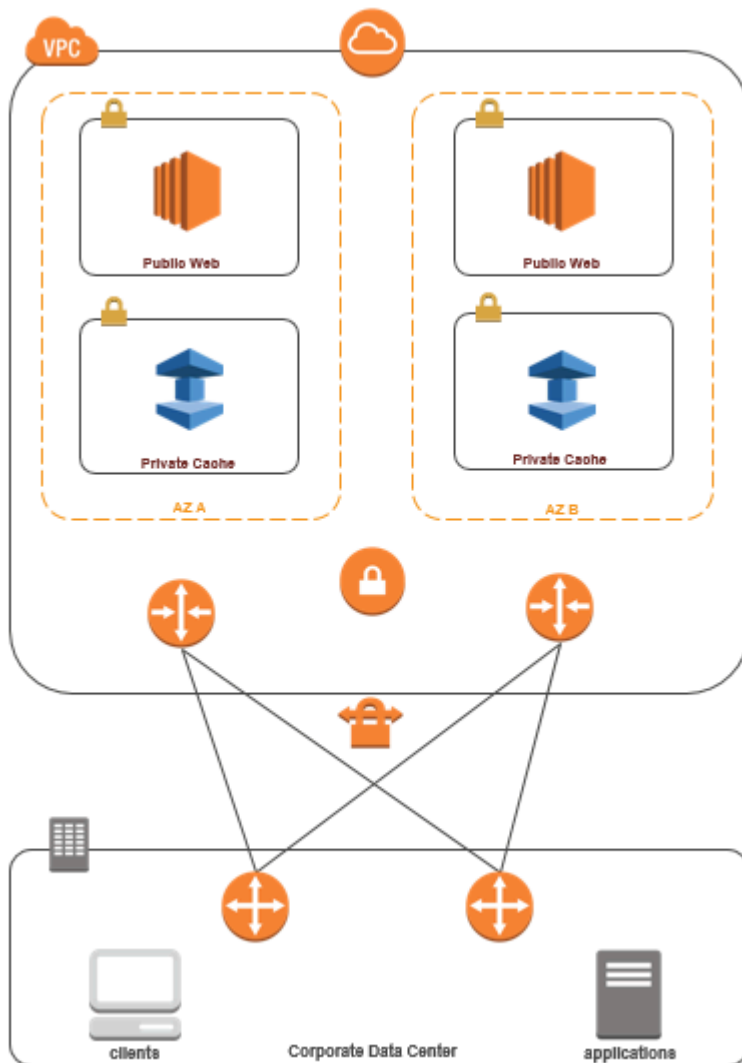
Un altro scenario possibile è un'architettura ibrida in cui i client o le applicazioni del data center del cliente potrebbero dover accedere a una ElastiCache cache nelVPC. Questo scenario è supportato anche a condizione che vi sia connettività tra i clienti VPC e il data center tramite VPN o Direct Connect.

### Argomenti

- [Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente utilizzando VPN la connettività](#)
- [Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect](#)

Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente utilizzando VPN la connettività

Il diagramma seguente illustra l'accesso a una ElastiCache cache da un'applicazione in esecuzione nella rete aziendale tramite VPN connessioni.



## Connessione ElastiCache dal data center tramite un VPN

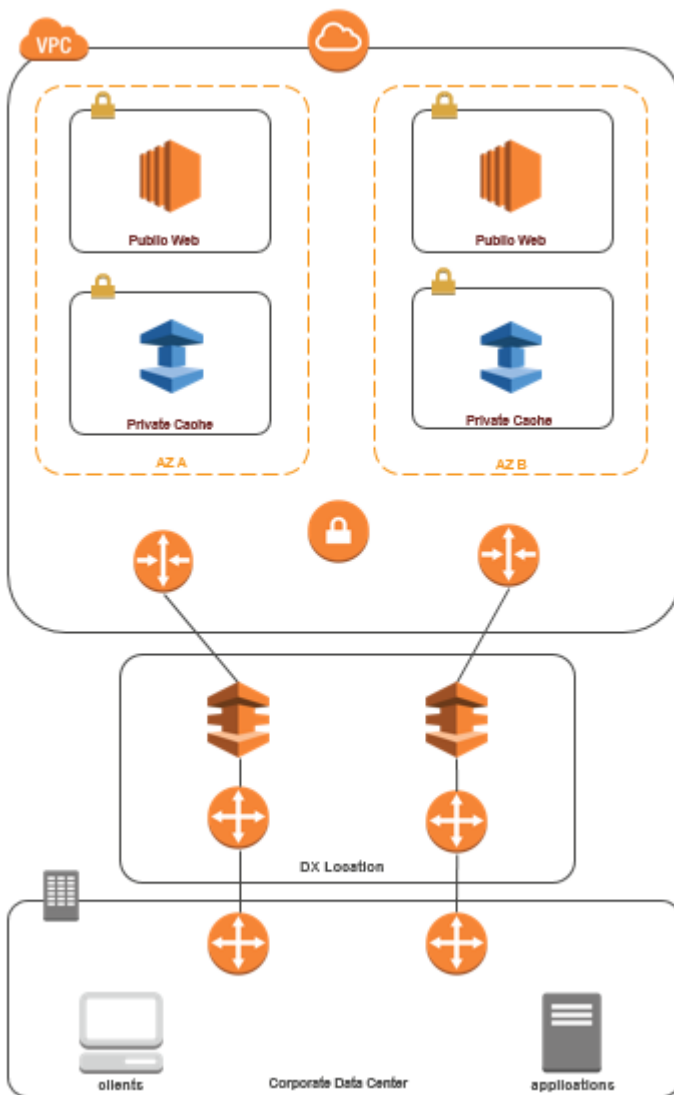
Per accedere a una cache in VPC un'applicazione locale tramite connessione VPN

1. Stabilisci la VPN connettività aggiungendo un gateway privato virtuale hardware al tuo VPC. Per ulteriori informazioni, consulta [Aggiungere un gateway privato virtuale hardware al tuo VPC](#).
2. Aggiorna la tabella VPC di routing per la sottorete in cui è distribuita la ElastiCache cache per consentire il traffico proveniente dal server delle applicazioni locale. In caso di routing dinamico, le BGP tue rotte potrebbero essere propagate automaticamente.
3. Modifica il gruppo di sicurezza della ElastiCache cache per consentire la connessione in entrata dai server delle applicazioni locali.

L'accesso a una cache tramite una VPN connessione introdurrà latenze di rete e costi aggiuntivi per il trasferimento dei dati.

Accesso a una ElastiCache cache da un'applicazione in esecuzione nel data center di un cliente tramite Direct Connect

Il diagramma seguente illustra l'accesso a una ElastiCache cache da un'applicazione in esecuzione sulla rete aziendale tramite Direct Connect.



Connessione ElastiCache dal data center tramite Direct Connect

Per accedere a una ElastiCache cache da un'applicazione in esecuzione nella rete utilizzando Direct Connect

1. Stabilire una connessione Direct Connect. Per ulteriori informazioni, consulta [Guida introduttiva a AWS Direct Connect](#).
2. Modifica il gruppo di sicurezza della ElastiCache cache per consentire la connessione in entrata dai server delle applicazioni locali.

L'accesso a una cache con una connessione DX può introdurre latenze di rete e ulteriori costi di trasferimento dei dati.



## Creazione di un cloud privato virtuale (VPC)

In questo esempio, crei un Amazon VPC con una sottorete privata per ogni zona di disponibilità.

### Creare un Amazon VPC (Console)

1. Accedi alla Console di AWS gestione e apri la VPC console Amazon all'indirizzo <https://console.aws.amazon.com/vpc/>.
2. Nella VPC dashboard, scegli Crea VPC.
3. In Risorse da creare, scegli VPCe altro ancora.
4. In Numero di zone di disponibilità (AZs), scegli il numero di zone di disponibilità in cui desideri avviare le sottoreti.
5. In Numero di sottoreti pubbliche, scegli il numero di sottoreti pubbliche che desideri aggiungere alle tue VPC.
6. In Numero di sottoreti private, scegli il numero di sottoreti private che desideri aggiungere alle tue VPC.

#### Tip

Prendi nota degli identificatori di sottorete, specificando quello pubblico e quello privato. Queste informazioni ti serviranno in seguito, quando lancerai i tuoi cluster e aggiungerai un'EC2istanza Amazon ad AmazonVPC.

7. Crea un gruppo VPC di sicurezza Amazon. Utilizzerai questo gruppo per il tuo cluster di cache e la tua EC2 istanza Amazon.
  - a. Nel pannello di navigazione della console di VPC gestione Amazon, scegli 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 VPCcasella, scegli l'identificatore per il tuo AmazonVPC.

**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.

**Outbound rules** [Info](#)

Type	Protocol	Port range	Destination	Description - optional	
All traffic	All	All	Custom	<input type="text" value=""/>	<input type="button" value="Delete"/>
			<input type="text" value="0.0.0.0"/>		

- 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 consentirà di connetterti alla tua EC2 istanza Amazon utilizzando Secure Shell (SSH).
    - 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 Crea una nuova regola, scegli SSH, quindi scegli Aggiungi regola.
    - d. Imposta i seguenti valori per la nuova regola in entrata per consentire HTTP l'accesso:
      - Tipo: HTTP
      - Fonte: 0.0.0.0/0

Scegliere Apply Rule Changes (Applica modifiche della regola).

Ora sei pronto per creare un gruppo di sottoreti di cache e avviare un cluster di cache nel tuo AmazonVPC.

- [Creazione di un gruppo di sottoreti](#)
- [Creazione di un cluster Memcached \(console\)](#).
- [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#).

## Connessione a una cache in esecuzione su Amazon VPC

Questo esempio mostra come avviare un'EC2istanza Amazon su AmazonVPC. Puoi quindi accedere a questa istanza e accedere alla ElastiCache cache in esecuzione su AmazonVPC.

### Connessione a una cache in esecuzione su Amazon VPC (Console)

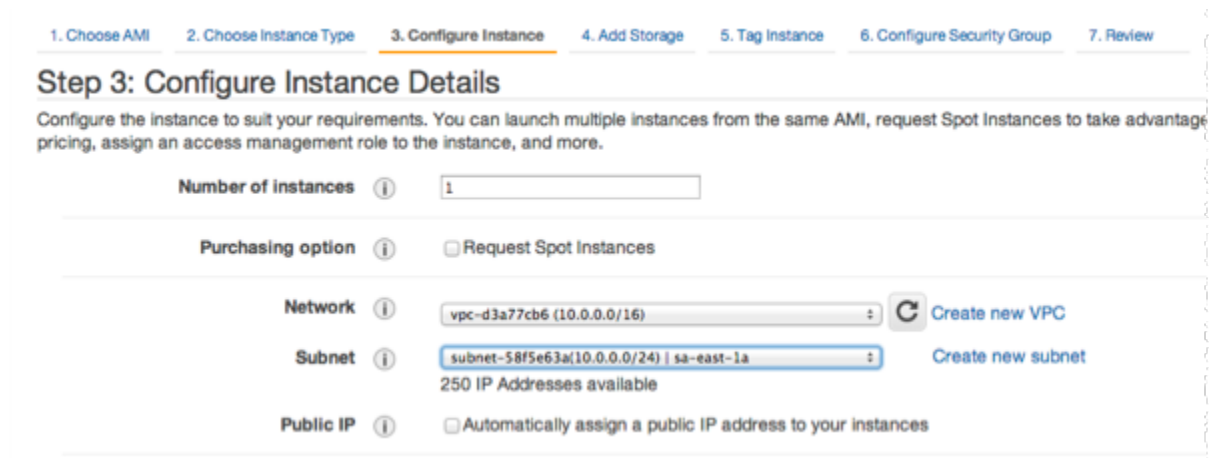
In questo esempio, crei un'EC2istanza Amazon nel tuo AmazonVPC. Puoi usare questa EC2 istanza Amazon per connetterti ai nodi di cache in esecuzione su AmazonVPC.

#### Note

Per informazioni sull'uso di AmazonEC2, consulta la [Amazon EC2 Getting Started Guide](#) nella [EC2documentazione di Amazon](#).

Per creare un'EC2istanza Amazon in Amazon VPC utilizzando la EC2 console Amazon

1. Accedi a AWS Management Console e apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nella console, scegliere Avvia istanza e attenersi alla seguente procedura:
3. Nella pagina Choose an Amazon Machine Image (AMI), scegli Amazon Linux a 64 bitAMI, quindi scegli 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 AmazonVPC.
  - 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 Instance, digita un nome per la tua EC2 istanza Amazon, quindi scegli 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 [Gruppi EC2 di sicurezza Amazon per istanze Linux](#).



- Scegli il nome del tuo gruppo di VPC sicurezza Amazon, quindi scegli Review and Launch.
- 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, consulta la [Amazon EC2 Getting Started Guide](#).

- Quando sei pronto per avviare la tua EC2 istanza Amazon, scegli Launch.

Ora puoi assegnare un indirizzo IP elastico all'EC2istanza Amazon appena creata. È necessario utilizzare questo indirizzo IP per connettersi all'EC2istanza Amazon.

Per assegnare un indirizzo IP elastico (Console)

- Apri la VPC console Amazon all'indirizzo <https://console.aws.amazon.com/vpc/>.

2. Nell'elenco di navigazione, scegli Elastic IPs.
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 finestra di dialogo Indirizzo associato, nella casella Istanza, scegli l'ID dell'EC2istanza Amazon che hai avviato.


Nella casella Indirizzo IP privato scegliere la casella per ottenere l'indirizzo IP privato, quindi scegliere Associa.

Ora puoi utilizzarlo SSH per connetterti all'EC2istanza Amazon utilizzando l'indirizzo IP elastico che hai creato.

Per connetterti alla tua EC2 istanza Amazon

- 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 disconnetterti ancora dalla tua EC2 istanza Amazon.

Ora sei pronto per interagire con il tuo ElastiCache cluster. 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)

- 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!
```

Ora puoi connetterti a un VPC con Memcached o Redis.

### Connessione a un con Memcached VPC

1. Accedi alla ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/> e ottieni l'endpoint per uno dei nodi del tuo cluster di cache. Per maggiori informazioni, consulta [Trovare gli endpoint di connessione](#).
2. Utilizzare telnet per connettersi all'endpoint del nodo della cache sulla porta 11211. 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 11211
```

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:

```
add mykey 0 3600 28
This is the value for mykey
```

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:

```
VALUE mykey 0 28
This is the value for my key
END
```

Per disconnettersi dal motore della cache, digitare quanto segue:

```
quit
```

## Connessione a un VPC con Redis

1. Vai alla ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/> e ottieni l'endpoint per uno dei nodi del tuo cluster di cache. Per ulteriori informazioni, consulta [Finding connection endpoint for Redis](#).
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. Vai alla ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/> e ottieni l'endpoint per uno dei nodi del tuo cluster di cache. Per ulteriori informazioni, consulta [Finding connection endpoint for Redis](#). OSS
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 di incorrere in costi aggiuntivi sul tuo AWS account, assicurati di eliminare tutte le risorse AWS che non desideri più dopo aver provato questi esempi.

## ElastiCache API e VPC endpoint di interfaccia (AWS PrivateLink)

Puoi stabilire una connessione privata tra i tuoi endpoint VPC e Amazon creando un ElastiCache API endpoint di interfaccia VPC. Gli endpoint di interfaccia sono alimentati da [AWS PrivateLink](#). AWS PrivateLink ti consente di accedere in modo privato alle ElastiCache API operazioni di Amazon senza un gateway Internet, un NAT dispositivo, una connessione o una VPN connessione AWS Direct Connect.

Le istanze nel tuo paese VPC non necessitano di indirizzi IP pubblici per comunicare con gli ElastiCache API endpoint Amazon. Inoltre, le tue istanze non necessitano di indirizzi IP pubblici per utilizzare nessuna delle operazioni disponibili. ElastiCache API Il traffico tra te VPC e Amazon ElastiCache non esce dalla rete Amazon. Ogni endpoint di interfaccia è rappresentato da una o più interfacce di rete elastiche nelle sottoreti. Per ulteriori informazioni sulle interfacce di rete elastiche, consulta Interfacce di [rete elastiche](#) nella Amazon EC2 User Guide.

- Per ulteriori informazioni sugli VPC endpoint, consulta [Interface VPC endpoints \(AWS PrivateLink\)](#) nella Amazon VPC User Guide.
- [Per ulteriori informazioni sulle ElastiCache API operazioni, consulta ElastiCache API operations.](#)

Dopo aver creato un VPC endpoint di interfaccia, se abiliti DNS i nomi host [privati](#) per l'endpoint, l'endpoint predefinito ElastiCache (<https://elasticache.Region.amazonaws.com>) si risolve sul tuo

endpoint. VPC Se non abiliti DNS i nomi host privati, Amazon VPC fornisce un nome di DNS endpoint che puoi utilizzare nel seguente formato:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Per ulteriori informazioni, consulta [Interface VPC Endpoints \(AWS PrivateLink\)](#) nella Amazon VPC User Guide. ElastiCache supporta l'effettuazione di chiamate a tutte le sue [APIazioni](#) all'interno del tuoVPC.

#### Note

DNSI nomi host privati possono essere abilitati per un solo VPC endpoint in. VPC Se desideri creare un VPC endpoint aggiuntivo, il DNS nome host privato deve essere disabilitato.

## Considerazioni per gli endpoint VPC

Prima di configurare un endpoint di interfaccia per gli VPC endpoint Amazon ElastiCache API, assicurati di esaminare le [proprietà e le limitazioni degli endpoint dell'interfaccia](#) nella Amazon VPC User Guide. Tutte le ElastiCache API operazioni relative alla gestione ElastiCache delle risorse Amazon sono disponibili presso il tuo VPC utilizzo AWS PrivateLink.

VPCle politiche degli endpoint sono supportate per gli ElastiCache API endpoint. Per impostazione predefinita, l'accesso completo alle ElastiCache API operazioni è consentito tramite l'endpoint. Per ulteriori informazioni, consulta [Controllare l'accesso ai servizi con VPC endpoint](#) nella Amazon VPC User Guide.

## Creazione di un VPC endpoint di interfaccia per ElastiCache API

Puoi creare un VPC endpoint per Amazon ElastiCache API utilizzando la VPC console Amazon o il AWS CLI. Per ulteriori informazioni, consulta [Creazione di un endpoint di interfaccia](#) nella Amazon VPC User Guide.

Dopo aver creato un VPC endpoint di interfaccia, puoi abilitare i DNS nomi host privati per l'endpoint. Quando lo fai, l' ElastiCache endpoint Amazon predefinito (`https://elasticache.Region.amazonaws.com`) si risolve nel tuo endpoint. VPC Per le AWS regioni Cina (Pechino) e Cina (Ningxia), puoi effettuare API richieste con l'endpoint utilizzando `for Beijing` e `Ningxia`. VPC `elasticache.cn-north-1.amazonaws.com.cn` `elasticache.cn-`

northwest-1.amazonaws.com.cn Per ulteriori informazioni, consulta [Accedere a un servizio tramite un endpoint di interfaccia](#) nella Amazon VPC User Guide.

## Creazione di una policy VPC sugli endpoint per Amazon ElastiCache API

Puoi allegare una policy sugli endpoint al tuo VPC endpoint che controlli l'accesso a ElastiCache API. La policy specifica quanto segue:

- Il principale che può eseguire azioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire azioni.

Per ulteriori informazioni, consulta [Controllare l'accesso ai servizi con VPC endpoint](#) nella Amazon VPC User Guide.

### Example VPC politica degli endpoint per ElastiCache API le azioni con Valkey o Redis OSS

Di seguito è riportato un esempio di policy sugli endpoint per ElastiCache API. Se associata a un endpoint, questa politica consente l'accesso alle ElastiCache API azioni elencate per tutti i principali su tutte le risorse.

```
{
 "Statement": [{
 "Principal": "*",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:ModifyCacheCluster",
 "elasticache:CreateSnapshot"
],
 "Resource": "*"
 }]
}
```

### Example VPC politica degli endpoint per ElastiCache le azioni (Memcached) API

Di seguito è riportato un esempio di policy sugli endpoint per ElastiCache API. Se associata a un endpoint, questa politica consente l'accesso alle ElastiCache API azioni elencate per tutti i principali su tutte le risorse.

```
{
 "Statement": [{
 "Principal": "*",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:ModifyCacheCluster"
],
 "Resource": "*"
 }]
}
```

Example VPCpolitica dell'endpoint che nega tutti gli accessi da un account specifico AWS

La seguente politica sugli VPC endpoint nega l'account AWS **123456789012** tutti gli accessi alle risorse tramite 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 (in genere private) che puoi designare per i tuoi cluster progettati autonomamente in esecuzione in un ambiente Amazon Virtual Private Cloud (). VPC

Se crei un cluster progettato autonomamente in AmazonVPC, devi utilizzare un gruppo di sottoreti. ElastiCache utilizza quel gruppo di sottorete per scegliere una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi.

ElastiCache fornisce un gruppo di IPv4 sottoreti predefinito oppure è possibile scegliere di crearne uno nuovo. Infatti IPv6, è necessario creare un gruppo di sottoreti con un IPv6 CIDR blocco. Se si sceglie il dual stack, è necessario selezionare un tipo di Discovery IP, oppure IPv6. IPv4

ElastiCache Serverless non utilizza una risorsa di un gruppo di sottoreti e utilizza invece un elenco di sottoreti direttamente durante la creazione.

Questa sezione spiega come creare e sfruttare sottoreti e gruppi di sottoreti per gestire l'accesso alle risorse. ElastiCache

Per ulteriori informazioni sull'utilizzo dei gruppi di sottoreti in un VPC ambiente Amazon, consulta [Accesso al ElastiCache cluster o al gruppo di replica](#).

## Argomenti

- [Creazione di un gruppo di sottoreti](#)
- [Assegnazione di un gruppo di sottoreti a 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 potresti voler designare per le tue cache in un VPC. Quando si avvia una cache in un VPC, è necessario selezionare un gruppo di sottoreti di cache. Quindi ElastiCache utilizza quel gruppo di sottoreti di cache per assegnare gli indirizzi IP all'interno di quella sottorete a ciascun nodo di cache nella 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 scegli IPv4 come tipo di rete, sarà disponibile un gruppo di sottoreti predefinito oppure puoi scegliere di crearne uno nuovo. ElastiCache utilizza quel gruppo di sottoreti per scegliere una sottorete e gli indirizzi IP all'interno di quella sottorete da associare ai nodi. Se scegli dual-stack o IPv6, verrai indirizzato a creare dual-stack o sottoreti IPv6. Per ulteriori informazioni sui tipi di rete, consulta [Network type](#). [Per ulteriori informazioni, consulta Creare una sottorete in un VPC](#)

Le procedure seguenti mostrano come creare un gruppo di sottoreti chiamato `mysubnetgroup` (console) AWS CLI, the e. ElastiCache API

### 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 AWS di gestione e aprirla 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 VPCID, scegli il tuo AmazonVPC.

- d. Per impostazione predefinita sono selezionate tutte le sottoreti. Nel pannello Sottoreti selezionate, fai clic su Gestisci e seleziona le Zone IDs di disponibilità o le [Local Zones](#) e le tue sottoreti private, quindi scegli Scegli.
5. Nel messaggio di conferma visualizzato, scegliere Close (Chiudi).

Il nuovo gruppo di sottoreti viene visualizzato nell'elenco dei gruppi di sottoreti della console.

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 AWS CLI [create-cache-subnet-group](#).



## Assegnazione di un gruppo di sottoreti a una cache

Dopo aver creato un gruppo di sottoreti, puoi avviare una cache in AmazonVPC. Per ulteriori informazioni, consulta gli argomenti seguenti.

- Cluster Memcached: per avviare un cluster Memcached, consultare [Creazione di un cluster Memcached \(console\)](#). Nel passaggio 7.a (Impostazioni avanzate di Memcached), scegli un gruppo di sottoreti. VPC
- Cluster Valkey o Redis autonomo: per avviare un OSS cluster Valkey o Redis a nodo singolo, vedi. OSS [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#) Nel passaggio 7.a (Impostazioni Redis OSS avanzate), scegli un gruppo di sottoreti. VPC
- Gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) — Per avviare un gruppo di replica Valkey o Redis OSS (modalità cluster disabilitata) in un, vedere. VPC [Creazione da zero di un gruppo di replica Valkey o Redis OSS \(modalità cluster disabilitata\)](#) Nel passaggio 7.b (Impostazioni Redis OSS avanzate), scegli un gruppo di sottoreti. VPC
- Gruppo di replica Valkey o Redis OSS (modalità cluster abilitata) —. [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#) Nel passaggio 6.i (OSSImpostazioni Redis avanzate), scegli un gruppo di sottoreti. VPC

## Modifica di un gruppo di sottoreti

È possibile modificare la descrizione di un gruppo di sottoreti o modificare l'elenco delle sottoreti IDs associate 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 a AWS Management Console e aprire la console all' ElastiCache 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' AWS CLI argomento [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. Accedi a AWS Management Console e apri la ElastiCache console 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 il AWS CLI, richiamate 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' AWS CLI argomento [delete-cache-subnet-group](#).

# Identity and Access Management per Amazon ElastiCache

AWS Identity and Access Management (IAM) è un dispositivo Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. IAM gli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. ElastiCache IAM è un dispositivo Servizio AWS che puoi utilizzare senza costi aggiuntivi.

## Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come ElastiCache funziona Amazon con IAM](#)
- [Esempi di policy basate sull'identità per Amazon ElastiCache](#)
- [Risoluzione dei problemi relativi all' ElastiCache identità e all'accesso ad Amazon](#)
- [Controllo accessi](#)
- [Panoramica sulla gestione delle autorizzazioni di accesso alle risorse ElastiCache](#)

## Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che svolgi. ElastiCache

Utente del servizio: se utilizzi il ElastiCache servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più ElastiCache funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in ElastiCache, consulta [Risoluzione dei problemi relativi all' ElastiCache identità e all'accesso ad Amazon](#).

Amministratore del servizio: se sei responsabile delle ElastiCache risorse della tua azienda, probabilmente hai pieno accesso a ElastiCache. È tuo compito determinare a quali ElastiCache funzionalità e risorse devono accedere gli utenti del servizio. È quindi necessario inviare richieste

all'IAM amministratore per modificare le autorizzazioni degli utenti del servizio. Consulta le informazioni contenute in questa pagina per comprendere i concetti di base di IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con ElastiCache, consulta [Come ElastiCache funziona Amazon con IAM](#).

IAM amministratore: se sei un IAM amministratore, potresti voler conoscere i dettagli su come scrivere politiche a cui gestire l'accesso ElastiCache. Per visualizzare esempi di policy ElastiCache basate sull'identità che puoi utilizzare in IAM, consulta. [Esempi di policy basate sull'identità per Amazon ElastiCache](#)

## Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. È necessario autenticarsi (accedere a AWS) come Utente root dell'account AWS, come IAM utente o assumendo un ruolo. IAM

È possibile accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Quando accedi come identità federata, l'amministratore aveva precedentemente configurato la federazione delle identità utilizzando i ruoli. IAM Quando si accede AWS utilizzando la federazione, si assume indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. CLI Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Firmare AWS API le richieste](#) nella Guida per l'IAM utente.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'AWS IAM Identity Center utente e [Utilizzo dell'autenticazione a più fattori \(MFA\) AWS nella Guida per l'IAM utente](#).

## Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per l'elenco completo delle attività che richiedono l'accesso come utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'IAMutente.

## Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web AWS Directory Service, la directory Identity Center o qualsiasi utente che accede Servizi AWS utilizzando credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per informazioni su IAM Identity Center, vedi [Cos'è IAM Identity Center?](#) nella Guida AWS IAM Identity Center per l'utente.

## IAM users and groups

Un [IAMutente](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Laddove possibile, consigliamo di fare affidamento su credenziali temporanee anziché creare IAM utenti con credenziali a lungo termine come password e chiavi di accesso. Tuttavia, se hai casi d'uso specifici che richiedono credenziali a lungo termine con IAM gli utenti, ti consigliamo di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta [Ruotare regolarmente le chiavi di accesso per i casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente. IAM

Un [IAMgruppo](#) 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,

potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per IAM amministrare le risorse.

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 IAM utente \(anziché un ruolo\)](#) nella Guida per l'IAMutente.

## IAMruoli

Un [IAMruolo](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un IAM utente, ma non è associato a una persona specifica. È possibile assumere temporaneamente un IAM ruolo in AWS Management Console [cambiando ruolo](#). È possibile assumere un ruolo chiamando un' AWS APIoperazione AWS CLI or o utilizzando un'operazione personalizzataURL. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere [Metodi per assumere un ruolo](#) nella Guida per l'IAMutente.

IAMI ruoli 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 informazioni sui ruoli per la federazione, vedere [Creazione di un ruolo per un provider di identità di terze parti](#) nella Guida per l'IAMutente. Se utilizzi IAM Identity Center, configuri un set di autorizzazioni. Per controllare a cosa possono accedere le identità dopo l'autenticazione, IAM Identity Center correla il set di autorizzazioni a un ruolo in. IAM Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .
- **Autorizzazioni IAM utente temporanee:** un IAM utente o un ruolo può assumere il IAM ruolo di assumere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso su più account:** puoi utilizzare un IAM ruolo per consentire a qualcuno (un responsabile fidato) di un altro account di accedere alle risorse del tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la [sezione Accesso alle risorse su più account IAM nella Guida per l'utente](#). IAM
- **Accesso tra servizi:** alcuni Servizi AWS utilizzano funzionalità in altri. Servizi AWS Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni



in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.

- **Sessioni di accesso diretto (FAS):** quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. FAS le richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [IAM ruolo](#) che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'interno IAM. Per ulteriori informazioni, vedere [Creazione di un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAM utente](#).
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.
- **Applicazioni in esecuzione su Amazon EC2:** puoi utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'EC2 istanza e che effettuano AWS CLI o effettuano AWS API richieste. È preferibile archiviare le chiavi di accesso all'interno dell'EC2 istanza. Per assegnare un AWS ruolo a un'EC2 istanza e renderlo disponibile per tutte le sue applicazioni, crea un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull'EC2 istanza di ottenere credenziali temporanee. Per ulteriori informazioni, consulta [Usare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida per l'IAM utente](#).

Per sapere se utilizzare IAM ruoli o IAM utenti, consulta [Quando creare un IAM ruolo \(anziché un utente\)](#) nella Guida per l'IAM utente.

## Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni.

AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come JSON documenti. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle JSON politiche, vedere [Panoramica delle JSON politiche](#) nella Guida per l'IAMutente.

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

IAMle politiche definiscono le autorizzazioni per un'azione indipendentemente dal metodo utilizzato per eseguire l'operazione. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale criterio può ottenere informazioni sul ruolo da AWS Management Console, da o da. AWS CLI AWS API

## Policy basate su identità

I criteri basati sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. [Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM IAM](#)

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli all'interno del tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una politica gestita o una politica in linea, consulta [Scelta tra politiche gestite e politiche in linea nella Guida](#) per l'IAMutente.

## Policy basate su risorse

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di politiche basate sulle risorse sono le policy di trust dei IAM ruoli e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli

per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite contenute IAM in una policy basata sulle risorse.

## Elenchi di controllo degli accessi ( ) ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica di Access control list \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

## Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità (utente o ruolo). IAM 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 nella Guida per l'utente. IAM IAM](#)
- **Politiche di controllo del servizio (SCPs):** SCPs sono JSON politiche che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. SCP Limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna Utente root dell'account AWS. Per ulteriori informazioni su Organizations and SCPs, consulta [le politiche di controllo dei servizi](#) nella Guida AWS Organizations per l'utente.

- **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 [le politiche di sessione](#) nella Guida IAM per l'utente.

## 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 determinare se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle politiche](#) nella Guida per l'IAMutente.

## Come ElastiCache funziona Amazon con IAM

Prima di IAM utilizzarlo per gestire l'accesso a ElastiCache, scopri con quali IAM funzionalità è disponibile l'uso ElastiCache.

### IAMfunzionalità che puoi usare con Amazon ElastiCache

IAMcaratteristica	ElastiCache supporto
<a href="#">Policy basate su identità</a>	Sì
<a href="#">Policy basate su risorse</a>	No
<a href="#">Azioni di policy</a>	Sì
<a href="#">Risorse relative alle policy</a>	Sì
<a href="#">Chiavi di condizione delle policy</a>	Sì
<a href="#">ACLs</a>	Sì
<a href="#">ABAC(tag nelle politiche)</a>	Sì
<a href="#">Credenziali temporanee</a>	Sì

IAMcaratteristica	ElastiCache supporto
<a href="#">Autorizzazioni del principale</a>	Sì
<a href="#">Ruoli di servizio</a>	Sì
<a href="#">Ruoli collegati al servizio</a>	Sì

Per avere una panoramica generale del funzionamento ElastiCache e degli altri AWS servizi con la maggior parte delle IAM funzionalità, consulta [AWS i servizi che funzionano con](#) la maggior parte delle funzionalità IAM nella Guida per l'IAMutente.

## Politiche basate sull'identità per ElastiCache

Supporta le policy basate su identità: sì

Le politiche basate sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. [Per informazioni su come creare una politica basata sull'identità, consulta Creazione di politiche nella Guida per l'utente. IAM IAM](#)

Con le politiche IAM basate sull'identità, puoi specificare azioni e risorse consentite o negate, nonché le condizioni in base alle quali le azioni sono consentite o negate. 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 ulteriori informazioni su tutti gli elementi che è possibile utilizzare in una JSON politica, vedere il [riferimento agli elementi IAM JSON della politica](#) nella Guida per l'IAMutente.

Esempi di policy basate su identità per ElastiCache

Per visualizzare esempi di politiche ElastiCache basate sull'identità, vedere. [Esempi di policy basate sull'identità per Amazon ElastiCache](#)

## Policy basate su risorse all'interno di ElastiCache

Supporta le policy basate su risorse: no

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di politiche basate sulle risorse sono le policy di trust dei IAM ruoli e le policy dei bucket di Amazon S3.

Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per abilitare l'accesso tra più account, puoi specificare un intero account o IAM entità in un altro account come principale in una politica basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un IAM amministratore dell'account fidato deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta la sezione [Cross Account Resource Access IAM nella Guida IAM per l'utente](#).

## Azioni politiche per ElastiCache

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare AWS JSON le policy per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'Actionelemento di una JSON policy descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni politiche in genere hanno lo stesso nome dell' AWS APIoperazione associata. Esistono alcune eccezioni, come le azioni basate solo sulle autorizzazioni che non hanno un'operazione corrispondente. API 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 ElastiCache azioni, consulta [Actions Defined by Amazon ElastiCache](#) nel Service Authorization Reference.

Le azioni politiche in ElastiCache uso utilizzano il seguente prefisso prima dell'azione:

```
elasticache
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
 "elasticache:action1",
 "elasticache:action2"
]
```

È possibile specificare più azioni tramite caratteri jolly (\*). Ad esempio, per specificare tutte le azioni che iniziano con la parola Describe, includi la seguente azione:

```
"Action": "elasticache:Describe*"
```

Per visualizzare esempi di politiche ElastiCache basate sull'identità, vedere [Esempi di policy basate sull'identità per Amazon ElastiCache](#)

## Risorse politiche per ElastiCache

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare AWS JSON le policy per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento Resource JSON policy specifica l'oggetto o gli oggetti a cui si applica l'azione. Le istruzioni devono includere un elemento Resource o un elemento NotResource. Come best practice, specifica una risorsa utilizzando il relativo [Amazon Resource Name \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (\*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di ElastiCache risorse e relativi ARNs, consulta [Resources Defined by Amazon ElastiCache](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare le caratteristiche ARN di ogni risorsa, consulta [Azioni definite da Amazon ElastiCache](#).

Per visualizzare esempi di politiche ElastiCache basate sull'identità, consulta [Esempi di policy basate sull'identità per Amazon ElastiCache](#)

## Chiavi di condizione delle policy per ElastiCache

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare AWS JSON le policy per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `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 si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica OR. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile concedere a un IAM utente l'autorizzazione ad accedere a una risorsa solo se è contrassegnata con il suo nome IAM utente. Per ulteriori informazioni, consulta [gli elementi IAM della politica: variabili e tag](#) nella Guida IAM per l'utente.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'IAM utente.

Per visualizzare un elenco di chiavi di ElastiCache condizione, consulta [Condition Keys for Amazon ElastiCache](#) nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Actions Defined by Amazon ElastiCache](#).

Per visualizzare esempi di politiche ElastiCache basate sull'identità, consulta [Esempi di policy basate sull'identità per Amazon ElastiCache](#)

## Liste di controllo degli accessi (ACL) in ACLs ElastiCache

Supporti ACLs: Sì

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON



## Controllo degli accessi basato sugli attributi () con ABAC ElastiCache

Supporti ABAC (tag nelle politiche): Sì

Il controllo degli accessi basato sugli attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. È possibile allegare tag a IAM entità (utenti o ruoli) e a molte AWS risorse. L'etichettatura di entità e risorse è il primo passo di ABAC. Quindi si progettano ABAC politiche per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa a cui sta tentando di accedere.

ABAC è utile in ambienti in rapida crescita e aiuta in situazioni in cui la gestione delle politiche diventa complicata.

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, vedere [Cos'è? ABAC](#) nella Guida IAM per l'utente. Per visualizzare un tutorial con i passaggi per la configurazione ABAC, consulta [Utilizzare il controllo di accesso basato sugli attributi \(ABAC\)](#) nella Guida per l'IAM utente.

## Utilizzo di credenziali temporanee con ElastiCache

Supporta le credenziali temporanee: sì

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione [Servizi AWS relativa alla funzionalità IAM nella Guida](#) per l'IAM utente.

Si utilizzano credenziali temporanee se si accede AWS Management Console utilizzando qualsiasi metodo tranne il nome utente e la password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-on (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sul cambio di ruolo, consulta [Passare a un ruolo \(console\)](#) nella Guida per l'IAM utente.

È possibile creare manualmente credenziali temporanee utilizzando AWS CLI o AWS API. È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, vedere [Credenziali di sicurezza temporanee](#) in IAM.

## Autorizzazioni del principale tra servizi per ElastiCache

Supporta sessioni di accesso diretto (FAS): Sì

Quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. FAS le richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta [Forward access sessions](#).

## Ruoli di servizio per ElastiCache

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un [IAM ruolo](#) che un servizio assume per eseguire azioni per conto dell'utente. Un IAM amministratore può creare, modificare ed eliminare un ruolo di servizio dall'interno IAM. Per ulteriori informazioni, vedere [Creazione di un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAM utente](#).

### Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità. ElastiCache Modifica i ruoli di servizio solo quando viene ElastiCache fornita una guida in tal senso.

## Ruoli collegati ai servizi per ElastiCache

Supporta ruoli collegati ai servizi: Sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono

visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un IAM amministratore può visualizzare, ma non modificare le autorizzazioni per i ruoli collegati al servizio.

[Per informazioni dettagliate sulla creazione o la gestione di ruoli collegati ai servizi, consulta AWS Servizi compatibili con. 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 sull'identità per Amazon ElastiCache

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare risorse. ElastiCache Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS API. Per concedere agli utenti il permesso di eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM policy. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

Per informazioni su come creare una politica IAM basata sull'identità utilizzando questi documenti di esempioJSON, consulta [Creazione di IAM politiche](#) nella Guida per l'IAMutente.

Per dettagli sulle azioni e sui tipi di risorse definiti da ElastiCache, incluso il formato di ARNs per ogni tipo di risorsa, consulta [Actions, Resources and Condition Keys per Amazon ElastiCache](#) nel Service Authorization Reference.

### Argomenti

- [Best practice per le policy](#)
- [Utilizzo della ElastiCache console](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

## Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare ElastiCache risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti

consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [le politiche AWS gestite o le politiche AWS gestite per le funzioni lavorative](#) nella Guida per l'IAMutente.

- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le IAM politiche, concedi solo le autorizzazioni necessarie 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 per applicare le autorizzazioni, consulta [Politiche](#) e autorizzazioni nella Guida IAM per l'utente. IAM IAM
- Utilizza le condizioni nelle IAM politiche per limitare ulteriormente l'accesso: puoi aggiungere una condizione alle tue politiche per limitare l'accesso ad azioni e risorse. Ad esempio, puoi scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. È inoltre possibile utilizzare condizioni per concedere l'accesso alle azioni di servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta [Elementi IAM JSON della politica: Condizione](#) nella Guida IAM per l'utente.
- Usa IAM Access Analyzer per convalidare IAM le tue policy e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio delle IAM policy () e alle best practice. JSON IAM IAMAccess Analyzer fornisce più di 100 controlli delle politiche e consigli pratici per aiutarti a creare policy sicure e funzionali. Per ulteriori informazioni, vedere [Convalida delle policy di IAM Access Analyzer nella Guida per l'utente. IAM](#)
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede l'utilizzo di IAM utenti o di un utente root Account AWS, attiva questa opzione MFA per una maggiore sicurezza. Per richiedere MFA quando vengono richiamate API le operazioni, aggiungi MFA delle condizioni alle tue politiche. Per ulteriori informazioni, consulta [Configurazione dell'APIaccesso MFA protetto nella Guida](#) per l'IAMutente.

Per ulteriori informazioni sulle procedure consigliate inIAM, consulta la sezione [Procedure consigliate in materia di sicurezza IAM nella](#) Guida per l'IAMutente.

## Utilizzo della ElastiCache console

Per accedere alla ElastiCache console Amazon, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle ElastiCache risorse del 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 autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso il AWS CLI o il. AWS API Consenti invece l'accesso solo alle azioni che corrispondono all'APIoperazione che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano continuare a utilizzare la ElastiCache console, allega anche la policy ElastiCache ConsoleAccess o la policy ReadOnly AWS gestita alle entità. Per ulteriori informazioni, consulta [Aggiungere autorizzazioni a un utente](#) nella Guida per l'IAMutente.

## Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra come è possibile creare una politica che consenta IAM agli utenti di visualizzare le politiche in linea e gestite allegate alla loro identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando o a livello di codice. AWS CLI  
AWS API

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ViewOwnUserInfo",
 "Effect": "Allow",
 "Action": [
 "iam:GetUserPolicy",
 "iam:ListGroupForUser",
 "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' ElastiCache identità e all'accesso ad Amazon

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con ElastiCache e IAM.

### Argomenti

- [Non sono autorizzato a eseguire alcuna azione in ElastiCache](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne al mio AWS account di accedere alle mie ElastiCache risorse](#)

### Non sono autorizzato a eseguire alcuna azione in ElastiCache

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

Il seguente esempio di errore si verifica quando l'utente mateojackson prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *my-example-widget* fittizia, ma non dispone di autorizzazioni `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 policy per poter accedere alla risorsa *my-example-widget* utilizzando l'operazione `elasticache:GetWidget`.

## Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole` azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a ElastiCache.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un IAM utente denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in ElastiCache. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di assistenza, contatta AWS l'amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Voglio consentire a persone esterne al mio AWS account di accedere alle mie ElastiCache risorse

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 i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se ElastiCache supporta queste funzionalità, consulta [Come ElastiCache funziona Amazon con IAM](#)
- Per informazioni su Account AWS come fornire l'accesso alle risorse di tua proprietà, consulta [Fornire l'accesso a un IAM utente di un altro Account AWS utente di tua proprietà](#) nella Guida per l'IAMutente.

- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a persone Account AWS di proprietà di terzi](#) nella Guida per l'IAMutente.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso agli utenti autenticati esternamente \(federazione delle identità\)](#) nella Guida per l'IAMutente.
- Per conoscere la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle [risorse tra account nella Guida per l'utente](#). IAM IAM

## Controllo accessi

È possibile disporre di credenziali valide per autenticare le richieste, ma a meno che non si disponga delle autorizzazioni necessarie non è possibile creare o accedere alle risorse. ElastiCache Ad esempio, è necessario disporre delle autorizzazioni per creare un cluster. ElastiCache

Le seguenti sezioni descrivono come gestire le autorizzazioni per. ElastiCache Consigliamo di leggere prima la panoramica.

- [Panoramica sulla gestione delle autorizzazioni di accesso alle risorse ElastiCache](#)
- [Utilizzo di politiche \(IAMpolitiche\) basate sull'identità per Amazon ElastiCache](#)



# Panoramica sulla gestione delle autorizzazioni di accesso alle risorse ElastiCache

Ogni AWS risorsa è di proprietà di un AWS account e le autorizzazioni per creare o accedere a una risorsa sono regolate dalle politiche di autorizzazione. Un amministratore di account può associare politiche di autorizzazione alle IAM identità (ovvero utenti, gruppi e ruoli). Inoltre, Amazon supporta ElastiCache anche l'associazione di politiche di autorizzazione alle risorse.

## Note

Un amministratore account (o un utente amministratore) è un utente con privilegi di amministratore. Per ulteriori informazioni, consulta le [IAMBest Practice](#) nella Guida per l'IAMutente.

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 IAM tramite un provider di identità:

Crea un ruolo per la federazione delle identità. Segui le istruzioni riportate in [Creazione di un ruolo per un provider di identità di terze parti \(federazione\)](#) nella Guida per l'IAMutente.

- IAMutenti:

- Crea un ruolo che l'utente possa assumere. Segui le istruzioni riportate nella sezione [Creazione di un ruolo per un IAM utente](#) nella Guida per l'IAMutente.
- (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate in [Aggiungere autorizzazioni a un utente \(console\)](#) nella Guida per l'IAMutente.

## Argomenti

- [ElastiCache Risorse e operazioni Amazon](#)
- [Informazioni sulla proprietà delle risorse](#)
- [Gestione dell'accesso alle risorse](#)

- [AWS politiche gestite per Amazon ElastiCache](#)
- [Utilizzo di politiche \(IAMpolitiche\) basate sull'identità per Amazon ElastiCache](#)
- [Autorizzazioni a livello di risorsa](#)
- [Utilizzo delle chiavi di condizione](#)
- [Utilizzo di ruoli collegati ai servizi per Amazon ElastiCache](#)
- [ElastiCache API autorizzazioni: riferimento ad azioni, risorse e condizioni](#)

## ElastiCache Risorse e operazioni Amazon

Per visualizzare un elenco dei tipi di ElastiCache risorse e relativi ARNs, consulta [Resources Defined by Amazon ElastiCache](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare le caratteristiche ARN di ogni risorsa, consulta [Azioni definite da Amazon ElastiCache](#).

## Informazioni sulla proprietà delle risorse

Il proprietario della risorsa è l' AWS account che ha creato la risorsa. In altre parole, il proprietario della risorsa è l' AWS account dell'entità principale che autentica la richiesta che crea la risorsa. Un'entità principale può essere l'account root, un IAM utente o un IAM ruolo). Negli esempi seguenti viene illustrato il funzionamento:

- Supponiamo di utilizzare le credenziali dell'account root del proprio AWS account per creare un cluster di cache. In questo caso, il tuo AWS account è il proprietario della risorsa. In ElastiCache, la risorsa è il cluster di cache.
- Supponiamo di creare un IAM utente nel AWS proprio account e di concedere a quell'utente le autorizzazioni per creare un cluster di cache. In questo caso, l'utente può creare un cluster di cache. Tuttavia, il tuo AWS account, a cui appartiene l'utente, possiede la risorsa del cluster di cache.
- Supponiamo che tu crei un IAM ruolo nel tuo AWS account con le autorizzazioni per creare un cluster di cache. In questo caso, chiunque sia in grado di assumere il ruolo può creare un cluster di cache. Il tuo AWS account, a cui appartiene il ruolo, possiede la risorsa del 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

Questa sezione illustra l'utilizzo IAM nel contesto di Amazon ElastiCache. Non fornisce informazioni dettagliate sul IAM servizio. Per la IAM documentazione completa, vedi [Cos'è IAM?](#) nella Guida IAM per l'utente. Per informazioni sulla sintassi e le descrizioni delle IAM policy, vedere [AWS IAMPolicy Reference](#) nella Guida per l'IAM utente.

Le politiche associate a un'IAM identità sono denominate politiche basate sull'identità (politiche). IAM Le policy collegate a una risorsa vengono definite policy basate sulle risorse.

### Argomenti

- [Politiche basate sull'identità \(politiche\) IAM](#)
- [Specifica degli elementi delle policy: operazioni, effetti, risorse ed entità](#)
- [Specifica delle condizioni in una policy](#)

### Politiche basate sull'identità (politiche) IAM

È possibile allegare politiche alle identità. IAM Ad esempio, puoi eseguire le operazioni seguenti:

- Collegare una policy di autorizzazione a un utente o a un gruppo nell'account – Per assegnare le autorizzazioni un amministratore di account può utilizzare una policy di autorizzazione associata a un utente specifico. In questo caso, l'utente è autorizzato a creare una ElastiCache risorsa, ad esempio un cluster di cache, un gruppo di parametri o un gruppo di sicurezza.
- Associare una politica di autorizzazioni a un ruolo (concedere autorizzazioni per più account): è possibile allegare una politica di autorizzazioni basata sull'identità a un ruolo per concedere autorizzazioni su più account. IAM Ad esempio, l'amministratore dell'Account A può creare un ruolo per concedere autorizzazioni su più account a un altro account (ad esempio, Account B) o a un AWS servizio come segue: AWS
  1. L'amministratore dell'account A crea un IAM ruolo e attribuisce una politica di autorizzazioni al ruolo che concede le autorizzazioni sulle risorse nell'Account A.
  2. L'amministratore dell'account A attribuisce una policy di attendibilità al ruolo, identificando l'account B come il principale per tale ruolo.
  3. L'amministratore dell'Account B può quindi delegare le autorizzazioni per assumere il ruolo a qualsiasi utente dell'Account B. In questo modo gli utenti dell'Account B possono creare o accedere a risorse nell'Account A. In alcuni casi, potresti voler concedere a un AWS servizio le

autorizzazioni per assumere il ruolo. Per supportare tale approccio, l'entità principale nella policy di trust può anche essere un'entità principale di un servizio AWS .

Per ulteriori informazioni sull'utilizzo per IAM delegare le autorizzazioni, vedere [Gestione degli accessi](#) nella Guida per l'utente. IAM

Di seguito è riportato un esempio di politica che consente a un utente di eseguire l'DescribeCacheClustersazione per il proprio AWS account. ElastiCache supporta anche l'identificazione di risorse specifiche utilizzando la risorsa ARNs per API le azioni. (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'utilizzo di politiche basate sull'identità con ElastiCache, vedere. [Utilizzo di politiche \(IAMpolitiche\) basate sull'identità per Amazon ElastiCache](#) Per ulteriori informazioni su utenti, gruppi, ruoli e autorizzazioni, consulta [Identità \(utenti, gruppi e ruoli\)](#) nella Guida per l'utente. IAM

Specifica degli elementi delle policy: operazioni, effetti, risorse ed entità

Per ogni ElastiCache risorsa Amazon (vedi [ElastiCache Risorse e operazioni Amazon](#)), il servizio definisce una serie di API operazioni (vedi [Azioni](#)). Per concedere le autorizzazioni per queste API operazioni, ElastiCache definisce una serie di azioni che puoi specificare in una politica. Ad esempio, per la risorsa ElastiCache cluster, vengono definite le seguenti azioni:CreateCacheCluster,DeleteCacheCluster, eDescribeCacheCluster. L'esecuzione di un'APIoperazione può richiedere autorizzazioni per più di un'azione.

Di seguito sono elencati gli elementi di base di una policy:

- **Risorsa:** in una policy, utilizzi un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy. Per ulteriori informazioni, consulta [ElastiCache Risorse e operazioni Amazon](#).
- **Operazione:** utilizzi le parole chiave per identificare le operazioni sulla risorsa da permettere o rifiutare. Ad esempio, a seconda di quanto specificato `Effect`, `elasticache:CreateCacheCluster` autorizzazione consente o nega all'utente le autorizzazioni per eseguire l'operazione `Amazon ElastiCacheCreateCacheCluster`.
- **Effetto:** l'effetto prodotto quando l'utente richiede l'operazione specifica, ovvero un'autorizzazione o un rifiuto. `Allow` 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.
- **Principal:** nelle politiche basate sull'identità (IAM policy), l'utente a cui è associata la policy 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 sulle descrizioni delle IAM policy, consulta [AWS IAM Policy Reference](#) nella Guida per l'utente. IAM

Per una tabella che mostra tutte le ElastiCache API azioni di Amazon, consulta [ElastiCache API autorizzazioni: riferimento ad azioni, risorse e condizioni](#).

### Specifiche delle condizioni in una policy

Quando concedi le autorizzazioni, puoi utilizzare il linguaggio della IAM policy per specificare le condizioni in cui una politica deve avere effetto. Ad esempio, potresti decidere che una policy venga applicata solo dopo una data specifica. Per ulteriori informazioni sulla specificazione delle condizioni in un linguaggio di policy, consulta [Condition nella Guida](#) per l'IAM utente.

Per esprimere le condizioni è necessario utilizzare chiavi di condizione predefinite. Per utilizzare i tasti ElastiCache di condizione specifici, vedere [Utilizzo delle chiavi di condizione](#). Esistono tasti di condizione AWS-wide che è possibile utilizzare a seconda delle esigenze. Per un elenco completo delle chiavi AWS-wide, consulta [Available Keys for Conditions nella Guida per l'IAM utente](#).

## AWS politiche gestite per Amazon ElastiCache

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti 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 politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove API operazioni per i servizi esistenti.

Per ulteriori informazioni, consulta [le politiche AWS gestite](#) nella Guida IAM per l'utente.

### AWS politica gestita: ElastiCacheServiceRolePolicy

Non puoi collegarti ElastiCacheServiceRolePolicy alle tue IAM entità. Questa policy è associata a un ruolo collegato al servizio che consente di eseguire azioni ElastiCache per conto dell'utente.

Questa politica consente di ElastiCache gestire AWS le risorse per conto dell'utente nella misura necessaria per la gestione della cache:

- `ec2`— Gestisci le risorse EC2 di rete da collegare ai nodi di cache, inclusi gli VPC endpoint (per le cache serverless), le interfacce di rete elastiche (ENIs) (per cluster progettati autonomamente) e i gruppi di sicurezza.
- `cloudwatch`— Emetti dati metrici dal servizio a. CloudWatch
- `outposts`— Consenti la creazione di nodi di cache su AWS Outposts.

Puoi trovare la [ElastiCacheServiceRolePolicy](#) policy sulla IAM console e [ElastiCacheServiceRolePolicy](#) nella AWS Managed Policy Reference Guide.

## AWS politica gestita: AmazonElastiCacheFullAccess

Puoi allegare la AmazonElastiCacheFullAccess politica alle tue IAM identità.

Questa politica consente ai responsabili l'accesso completo all' ElastiCache utilizzo della Console di AWS gestione:

- `elasticache`— Accedi a tuttiAPIs.
- `iam`: crea il ruolo collegato ai servizi necessario per il funzionamento dei servizi.
- `ec2`— Descrivi EC2 le risorse dipendenti necessarie per la creazione della cache (VPC, sottorete, gruppo di sicurezza) e consenti la creazione di VPC endpoint (per le cache serverless).
- `kms`— Consenti l'utilizzo di un modulo gestito dal cliente. CMKs encryption-at-rest
- `cloudwatch`— Consenti l'accesso alle metriche per visualizzare le ElastiCache metriche 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 compilare SNS gli argomenti relativi alla funzionalità di notifica nella console.

È possibile trovare la [AmazonElastiCacheFullAccess](#) policy sulla IAM console e [AmazonElastiCacheFullAccess](#) nella AWS Managed Policy Reference Guide.

## AWS politica gestita: AmazonElastiCacheReadOnlyAccess

Puoi allegare la AmazonElastiCacheReadOnlyAccess politica alle tue IAM identità.

Questa politica consente ai principali l'accesso in sola lettura all' ElastiCache utilizzo della Console di gestione: AWS

- `elasticache`— Accesso in sola lettura. Describe APIs

È possibile trovare la [AmazonElastiCacheReadOnlyAccess](#) policy sulla IAM console e [AmazonElastiCacheReadOnlyAccess](#) nella AWS Managed Policy Reference Guide.

## ElastiCache aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite ElastiCache da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al RSS feed nella pagina della cronologia dei ElastiCache documenti.

Modifica	Descrizione	Data
<a href="#">AmazonElastiCacheFullAccess</a> : aggiornamento a una policy esistente	ElastiCache ha aggiunto nuove autorizzazioni per consentire la gestione delle cache serverless e per consentire l'utilizzo di tutte le funzionalità del servizio tramite la console.	27 novembre 2023
<a href="#">ElastiCacheServiceRolePolicy</a> : aggiornamento a una policy esistente	ElastiCache ha aggiunto nuove autorizzazioni per consentire la gestione degli VPC endpoint per le risorse di cache serverless.	27 novembre 2023
ElastiCache ha iniziato a tenere traccia delle modifiche	ElastiCache ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	7 febbraio 2020

## Utilizzo di politiche (IAMpolitiche) basate sull'identità per Amazon ElastiCache

Questo argomento fornisce esempi di politiche basate sull'identità in cui un amministratore di account può associare politiche di autorizzazione alle IAM identità (ovvero utenti, gruppi e ruoli).

### Important

Ti consigliamo di leggere prima gli argomenti che spiegano i concetti e le opzioni di base per gestire l'accesso alle ElastiCache risorse di Amazon. Per ulteriori informazioni, consulta [Panoramica sulla gestione delle autorizzazioni di accesso alle risorse ElastiCache](#).



In questa sezione vengono trattati gli argomenti seguenti:

- [AWS politiche gestite per Amazon ElastiCache](#)
- [Esempi di policy gestite dal cliente](#)

Di seguito viene mostrato un esempio di politica di autorizzazioni quando si utilizza RedisOSS.

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

Di seguito viene mostrato un esempio di politica di autorizzazioni quando si utilizza Memcached.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "AllowClusterPermissions",
 "Effect": "Allow",
 "Action": [
```

```

 "elasticache:CreateServerlessCache",
 "elasticache:CreateCacheCluster",
 "elasticache:DescribeServerlessCaches",
 "elasticache:DescribeCacheClusters",
 "elasticache:ModifyServerlessCache",
 "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 concede le autorizzazioni per le ElastiCache azioni Amazon (`elasticache:Create*`, `elasticache:Describe*`) `elasticache:Modify*`
- La seconda istruzione concede le autorizzazioni per l'IAM azione (`iam:PassRole`) sul nome del IAM ruolo 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 si associa una politica di autorizzazioni a un IAM ruolo, il principale identificato nella politica di fiducia del ruolo ottiene le autorizzazioni.

Per una tabella che mostra tutte le ElastiCache API azioni di Amazon e le risorse a cui si applicano, consulta [ElastiCache API autorizzazioni: riferimento ad azioni, 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: consentire a un utente di chiamare IAM CreateServiceLinkedRole API](#)). Oppure avresti dovuto creare un ruolo ElastiCache collegato al servizio.

Se combinate con le autorizzazioni minime necessarie per utilizzare la ElastiCache console Amazon, le politiche di esempio in questa sezione concedono autorizzazioni aggiuntive. Gli esempi sono rilevanti anche per il AWS SDKs e il. AWS CLI

Per istruzioni sulla configurazione di IAM utenti e gruppi, vedere [Creazione del primo gruppo di IAM utenti e amministratori](#) nella Guida per l'IAMutente.

#### Important

Verifica sempre accuratamente IAM le tue politiche prima di utilizzarle in produzione. Alcune ElastiCache azioni che sembrano semplici possono richiedere altre azioni per supportarle quando si utilizza la ElastiCache console. Ad esempio, `elasticache:CreateCacheCluster` concede le autorizzazioni per creare cluster di ElastiCache cache. Tuttavia, per eseguire questa operazione, la ElastiCache console utilizza una serie di `List` azioni `Describe` e per compilare gli elenchi delle console.

#### Esempi

- [Esempio 1: consentire a un utente l'accesso in sola lettura alle risorse ElastiCache](#)
- [Esempio 2: consentire a un utente di eseguire attività comuni di amministratore ElastiCache di sistema](#)
- [Esempio 3: consentire a un utente di accedere a tutte le azioni ElastiCache API](#)
- [Esempio 4: consentire a un utente di chiamare IAM `CreateServiceLinkedRole` API](#)
- [Esempio 5: consentire a un utente di connettersi alla cache serverless utilizzando l'autenticazione IAM](#)

#### Esempio 1: consentire a un utente l'accesso in sola lettura alle risorse ElastiCache

La seguente politica concede ElastiCache azioni di autorizzazione che consentono a un utente di elencare le risorse. In genere, si collega questo tipo di policy di autorizzazione a un gruppo di gestori.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECReadOnly",
 "Effect": "Allow",
 "Action": [
 "elasticache:Describe*",

```

```

 "elasticache:List*"],
 "Resource": "*"
 }
]
}

```

### Esempio 2: consentire a un utente di eseguire attività comuni di amministratore ElastiCache di sistema

Le attività comuni dell'amministratore di sistema includono la modifica delle risorse. Un amministratore di sistema può anche voler ottenere informazioni sugli ElastiCache eventi. La seguente politica concede a un utente le autorizzazioni per eseguire ElastiCache azioni per queste attività comuni dell'amministratore di sistema. In genere, si collega questo tipo di policy di autorizzazione al gruppo degli amministratori di sistema.

```

{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECAAllowMutations",
 "Effect": "Allow",
 "Action": [
 "elasticache:Modify*",
 "elasticache:Describe*",
 "elasticache:ResetCacheParameterGroup"
],
 "Resource": "*"
 }
]
}

```

### Esempio 3: consentire a un utente di accedere a tutte le azioni ElastiCache API

La seguente politica consente a un utente di accedere a tutte le ElastiCache azioni. 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: consentire a un utente di chiamare IAM CreateServiceLinkedRole API

La seguente politica consente all'utente di chiamare il IAM CreateServiceLinkedRoleAPI. Si consiglia di concedere questo tipo di politica di autorizzazione all'utente che richiama operazioni mutative. ElastiCache

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CreateSLRAllows",
 "Effect": "Allow",
 "Action": [
 "iam:CreateServiceLinkedRole"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "iam:AWS ServiceName": "elasticache.amazonaws.com"
 }
 }
 }
]
}

```

#### Esempio 5: consentire a un utente di connettersi alla cache serverless utilizzando l'autenticazione IAM

La seguente politica consente a qualsiasi utente di connettersi a qualsiasi cache serverless utilizzando l'IAMautenticazione 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 l'ambito delle autorizzazioni specificando le risorse in una IAM politica. Molte ElastiCache API azioni supportano un tipo di risorsa che varia a seconda del comportamento dell'azione. Ogni dichiarazione IAM politica concede l'autorizzazione a un'azione 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 API azioni, puoi limitare le risorse che un utente può modificare specificando l'Amazon Resource Name (ARN) di una risorsa o uno ARN schema che corrisponda a più risorse. Per limitare le autorizzazioni per risorsa, specifica la risorsa per. ARN

Per visualizzare un elenco dei tipi di ElastiCache risorse e relativi ARNs, consulta [Resources Defined by Amazon ElastiCache](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare le caratteristiche ARN di ogni risorsa, consulta [Azioni definite da Amazon ElastiCache](#).

### Esempi

- [Esempio 1: consenti a un utente l'accesso completo a tipi di ElastiCache risorse specifici](#)
- [Esempio 2: rifiutare a un utente l'accesso a una cache serverless](#)

## Esempio 1: consenti a un utente l'accesso completo a tipi di ElastiCache risorse specifici

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

È possibile specificare le condizioni che determinano l'effetto di una IAM politica. In ElastiCache, è possibile utilizzare l'Conditionelemento di una JSON politica per confrontare le chiavi nel contesto della richiesta con i valori chiave specificati nella politica. Per ulteriori informazioni, consulta [Elementi IAM JSON della politica: Condizione](#).

Per visualizzare un elenco di chiavi di ElastiCache condizione, consulta [Condition Keys for Amazon ElastiCache](#) nel Service Authorization Reference.

Per un elenco delle chiavi di condizione globali, consulta [Chiavi di contesto delle condizioni globali AWS](#).

## Specifica delle condizioni: Uso delle chiavi di condizione

Per implementare un controllo granulare, scrivi una politica di IAM autorizzazioni che specifichi le condizioni per controllare una serie di parametri individuali su determinate richieste. Quindi applichi la politica a IAM utenti, gruppi o ruoli creati utilizzando la console. IAM

Per applicare una condizione, aggiungi le informazioni sulla condizione all'informativa IAM sulla politica. Nell'esempio seguente viene specificata la condizione secondo la quale qualsiasi cluster di cache progettato autonomamente avrà nodi di tipo `cache.r5.large`.

Di seguito viene mostrato un esempio di questa politica di autorizzazioni quando si utilizza Valkey o Redis. OSS

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



```

 }
 }
]
}

```

Di seguito viene mostrato un esempio di questa politica di autorizzazioni quando si utilizza Memcached.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheNodeType": [
 "cache.r5.large"
]
 }
 }
 }
]
}

```

Per ulteriori informazioni, consulta [Taggare le tue risorse ElastiCache](#).

Per ulteriori informazioni sull'utilizzo degli operatori delle condizioni di policy, consulta [ElastiCache API autorizzazioni: riferimento ad azioni, risorse e condizioni](#).

Policy di esempio: Utilizzo di condizioni per il controllo granulare dei parametri

Questa sezione mostra alcuni esempi di policy per l'implementazione di un controllo granulare degli accessi sui parametri elencati in precedenza. ElastiCache

1. `elasticache:MaximumDataStorage`: Specificare la memorizzazione massima dei dati 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:m aximumECPUPer Second`: specifica il valore massimo 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 cache al secondo. ECPUs

```

{
 "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`: Specificate quali `NodeType` possono essere create da un utente. 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:CacheNodeType`: Con Memcached, specifica quali `NodeType` (i) 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"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheNodeType": [
 "cache.t2.micro",
 "cache.t2.medium"
]
 }
 }
 }
]
}

```

5. `elasticache:NumNodeGroups`: Crea 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"
 }
 }
}
]
}

```

#### 6. 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"
 }
 }
}
]
}

```

### 7. elasticache:EngineVersion: Specificare l'utilizzo della versione 5.0.6 del motore.

```

{
 "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"
 }
 }
 }
]
}

```

```

 }
 }
]
}

```

## 8. elasticache:EngineVersion: Specificare l'utilizzo della versione 1.6.6 del motore Memcached

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:EngineVersion": "1.6.6"
 }
 }
 }
]
}

```

## 9. elasticache:EngineType: Specificare l'utilizzo solo di un motore Valkey o Redis. OSS

```

{
 "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"
 }
 }
 }
]
}

```

**10elasticache:AtRestEncryptionEnabled:** Specificate che i gruppi di replica verranno 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"
 }
 }
}
]
}

```

## 11.elasticache: TransitEncryptionEnabled

- Imposta la chiave di `elasticache:TransitEncryptionEnabled` condizione su `false` per l'[CreateReplicationGroup](#) azione per specificare che i gruppi di replica possono essere creati solo quando non TLS vengono utilizzati:

```

{
 "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 `elasticache:TransitEncryptionEnabled` condizione è impostata su `false` in una politica per l'[CreateReplicationGroup](#) azione, una `CreateReplicationGroup` richiesta sarà consentita solo se non TLS viene utilizzata (ovvero se la richiesta non include un `TransitEncryptionEnabled` parametro impostato su `true` o un `TransitEncryptionMode` parametro impostato `required` su).

- b. Imposta la chiave `elasticache:TransitEncryptionEnabled` conditon su `true` per l'[CreateReplicationGroup](#) azione per specificare che i gruppi di replica possono essere creati solo quando TLS vengono utilizzati:

```

{
 "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 `elasticache:TransitEncryptionEnabled` condizione è impostata su `true` in una politica per l'[CreateReplicationGroup](#) azione, una `CreateReplicationGroup` richiesta sarà consentita solo se la richiesta include un `TransitEncryptionEnabled` parametro impostato su `true` e un `TransitEncryptionMode` parametro impostato su `required`

- c. Impostato su `elasticache:TransitEncryptionEnabled true` per specificare che i `ModifyReplicationGroup` gruppi di replica possono essere modificati solo quando TLS vengono utilizzati:

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:ModifyReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "BoolIfExists": {
 "elasticache:TransitEncryptionEnabled": "true"
 }
 }
 }
]
}

```

```

 }
]
}

```

Quando la chiave di `elasticache:TransitEncryptionEnabled` condizione è impostata su `true` in una politica per l'[ModifyReplicationGroup](#) azione, una `ModifyReplicationGroup` richiesta sarà consentita solo se la richiesta include un `TransitEncryptionMode` parametro impostato `required` su. Facoltativamente, è possibile includere anche il `TransitEncryptionEnabled` parametro impostato su, ma in questo caso non è necessario per abilitarlo. `true` TLS

12 `elasticache:AutomaticFailoverEnabled`: Specificate che i gruppi di replica verranno 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"
 }
 }
 }
]
}

```

```
]
}
```

13 `elasticache:MultiAZEnabled`: Specificate 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"
 }
 }
 }
]
}
```

14 `elasticache:ClusterModeEnabled`: Specificate 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"
 }
 }
 }
]
}

```

15 `elasticache:AuthTokenEnabled`: Specificate che i gruppi di replica possono essere creati solo con il token abilitato. AUTH

```

{
 "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"
 }
 }
 }
]
}

```

**elasticache:SnapshotRetentionLimit:** Specificate il numero di giorni (o min/max) per conservare l'istantanea. 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"
 }
 }
 }
]
}

```

### 17.elasticache:KmsKeyId: Specificate l'utilizzo delle chiavi gestite dal cliente. AWS KMS

```

{
 "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"
 }
 }
 }
]
}

```

18. `elasticache:CacheParameterGroupName`: Specificate un gruppo di parametri non predefinito con parametri specifici di un'organizzazione sui 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 limita l'uso del 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"
 }
 }
]
}

```

19 `elasticache:CacheParameterGroupName`: Con Memcached, specifica 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 limita l'utilizzo del solo "». `my-org-param-group`

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:parametergroup:*",
 "arn:aws:elasticache:*:*:subnetgroup:*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "elasticache:CacheParameterGroupName": "my-org-param-group"
 }
 }
 }
]
}

```

```

 }
]
}

```

## 20elasticache:CreateCacheCluster: Negare l'CreateCacheClusterazione se il tag di richiesta Project è mancante o non è uguale a, o. Dev QA 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"
]
 }
 }
]
}

```

21 `elasticache:CacheNodeType`: Consentire `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"
 }
}
]
}

```

## 22.elasticache:CacheNodeType: Permesso CreateCacheCluster con cacheNodeType cache.r5.large o cache.r6g.4xlarge e tag. Project=XYZ

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster"
],
 "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 AWS Identity and Access Management (IAM) ruoli [collegati ai servizi](#). Un ruolo collegato al servizio è un tipo di IAM ruolo unico collegato direttamente a un AWS servizio, come Amazon. ElastiCache I ruoli ElastiCache collegati ai servizi Amazon sono predefiniti da Amazon. ElastiCache Includono tutte le autorizzazioni necessarie al servizio per richiamare servizi AWS per conto dei cluster.

Un ruolo collegato al servizio semplifica la configurazione di Amazon ElastiCache perché non è necessario aggiungere manualmente le autorizzazioni necessarie. I ruoli esistono già nel tuo AWS account ma sono collegati ai casi ElastiCache d'uso di Amazon e dispongono di autorizzazioni predefinite. Solo Amazon ElastiCache può assumere questi ruoli e solo questi ruoli possono utilizzare la politica di autorizzazioni predefinita. È possibile eliminare i ruoli solo dopo aver eliminato le risorse correlate. In questo modo proteggi le tue ElastiCache risorse Amazon perché non puoi rimuovere inavvertitamente le autorizzazioni necessarie per accedere alle risorse.

Per informazioni su altri servizi che supportano ruoli collegati ai servizi, consulta [AWS Servizi con cui funziona IAM e cerca i servizi con](#) Sì nella colonna Ruolo collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

### Indice

- [Autorizzazioni relative ai ruoli collegati 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 \(console\) IAM](#)
  - [Creazione di un ruolo collegato al servizio \(\) IAM CLI](#)
  - [Creazione di un ruolo collegato ai servizi \(\) IAM API](#)
- [Modifica della descrizione di un ruolo collegato ai servizi per Amazon ElastiCache](#)
  - [Modifica della descrizione di un ruolo collegato ai servizi \(console\) IAM](#)
  - [Modifica della descrizione di un ruolo collegato al servizio \(\) IAM CLI](#)
  - [Modifica della descrizione di un ruolo collegato al servizio \(\) IAM API](#)
- [Eliminazione di un ruolo collegato ai servizi per Amazon ElastiCache](#)
  - [Pulizia di un ruolo collegato ai servizi](#)
  - [Eliminazione di un ruolo collegato ai servizi \(console\) IAM](#)
  - [Eliminazione di un ruolo collegato al servizio \(\) IAM CLI](#)
  - [Eliminazione di un ruolo collegato al servizio \(\) IAM API](#)

Autorizzazioni relative ai ruoli collegati ai servizi per Amazon ElastiCache

Autorizzazioni per creare un ruolo collegato ai servizi

Per consentire a un'IAMentità di creare un ruolo collegato al servizio AWS  
ServiceRoleForElastiCache

Aggiungi la seguente dichiarazione politica alle autorizzazioni per quell'entità: IAM

```
{
 "Effect": "Allow",
 "Action": [
 "iam:CreateServiceLinkedRole",
 "iam:PutRolePolicy"
],
 "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS
ServiceRoleForElastiCache*",
 "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

Per consentire a un'IAMentità di eliminare un ruolo collegato al AWS ServiceRoleForElastiCache  
servizio



Aggiungi la seguente dichiarazione politica alle autorizzazioni per quell'entità: IAM

```
{
 "Effect": "Allow",
 "Action": [
 "iam:DeleteServiceLinkedRole",
 "iam:GetServiceLinkedRoleDeletionStatus"
],
 "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWS
ServiceRoleForElastiCache*",
 "Condition": {"StringLike": {"iam:AWS ServiceName": "elasticache.amazonaws.com"}}
}
```

In alternativa, puoi utilizzare una policy AWS gestita per fornire l'accesso completo ad Amazon ElastiCache.

Creazione di un ruolo collegato ai servizi () IAM

È possibile creare un ruolo collegato al servizio utilizzando la IAM console, oppure. CLI API

Creazione di un ruolo collegato ai servizi (console) IAM

È possibile utilizzare la IAM console per creare un ruolo collegato al servizio.

Come creare un ruolo collegato ai servizi (console)

1. Accedi AWS Management Console e apri la IAM console all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nel riquadro di navigazione della IAM console, scegli Ruoli. Quindi seleziona Create new role (Crea nuovo ruolo).
3. In Select type of trusted entity (Seleziona tipo di entità attendibile), scegli Service AWS .
4. In Oppure seleziona un servizio per visualizzarne i casi d'uso, scegli 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 al servizio ( ) IAM CLI

È possibile utilizzare IAM le operazioni di AWS Command Line Interface per creare un ruolo collegato al servizio. Questo ruolo può includere la policy di attendibilità e le policy inline che il servizio richiede per assumere il ruolo.

Per creare un ruolo collegato al servizio ( ) CLI

Attenersi alle operazioni seguenti:

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

### Creazione di un ruolo collegato ai servizi ( ) IAM API

È possibile utilizzare il IAM API per creare un ruolo collegato al servizio. 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 al servizio ( ) API

Usa il [CreateServiceLinkedRole](#) API chiamata. 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 AWS ServiceRoleForElastiCache collegato al servizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. Tuttavia, puoi modificare la descrizione del ruolo utilizzando IAM

### Modifica della descrizione di un ruolo collegato ai servizi (console) IAM

È possibile utilizzare la IAM console per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato ai servizi (console)

1. Nel riquadro di navigazione della IAM console, scegli Ruoli.
2. Scegliere il nome del ruolo da modificare.
3. Nella parte destra di Role description (Descrizione ruolo), scegliere Edit (Modifica).

4. Digita una nuova descrizione nella casella e scegli Save (Salva).

## Modifica della descrizione di un ruolo collegato al servizio () IAM CLI

È possibile utilizzare IAM le operazioni di AWS Command Line Interface per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato al servizio () CLI

1. (Facoltativo) Per visualizzare la descrizione corrente di un ruolo, utilizzate l'operazione AWS CLI forIAM. [get-role](#)

### Example

```
$ aws iam get-role --role-name AWS ServiceRoleForElastiCache
```

Utilizzate il nome del ruolo, non ilARN, per fare riferimento ai ruoli con le CLI operazioni. Ad esempio, se un ruolo presenta le seguenti caratteristicheARN:arn:aws:iam::123456789012:role/myrole, fate riferimento al ruolo comemyrole.

2. Per aggiornare la descrizione di un ruolo collegato al servizio, usa l'operazione AWS CLI forIAM. [update-role-description](#)

Per Linux, macOS o Unix:

```
$ aws iam update-role-description \
 --role-name AWS ServiceRoleForElastiCache \
 --description "new description"
```

Per Windows:

```
$ aws iam update-role-description ^\
 --role-name AWS ServiceRoleForElastiCache ^\
 --description "new description"
```

## Modifica della descrizione di un ruolo collegato al servizio () IAM API

È possibile utilizzare il IAM API per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato al servizio () API

1. (Facoltativo) Per visualizzare la descrizione corrente di un ruolo, utilizzate l'operazione IAM API [GetRole](#).

#### Example

```
https://iam.amazonaws.com/
?Action=GetRole
&RoleName=AWS ServiceRoleForElastiCache
&Version=2010-05-08
&AUTHPARAMS
```

2. Per aggiornare la descrizione di un ruolo, usa l'IAM API operazione [UpdateRoleDescription](#).

#### Example

```
https://iam.amazonaws.com/
?Action=UpdateRoleDescription
&RoleName=AWS ServiceRoleForElastiCache
&Version=2010-05-08
&Description="New description"
```

## Eliminazione di un ruolo collegato ai servizi per Amazon ElastiCache

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare.

Amazon ElastiCache non elimina il ruolo collegato al servizio per te.

### Pulizia di un ruolo collegato ai servizi

Prima di poter eliminare un ruolo collegato IAM al servizio, verifica innanzitutto che al ruolo non siano associate risorse (cluster o gruppi di replica).

Per verificare se il ruolo collegato al servizio ha una sessione attiva nella console IAM

1. Accedi a AWS Management Console e apri la IAM console all'indirizzo. <https://console.aws.amazon.com/iam/>

2. Nel riquadro di navigazione della IAM console, scegli Ruoli. Quindi scegli il nome (non la casella di controllo) del AWS ServiceRoleForElastiCache ruolo.
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, scegliere la scheda Access Advisor (Consulente accessi).
4. Nella scheda Access Advisor (Consulente accessi), esamina l'attività recente per il ruolo collegato ai servizi.

Per eliminare ElastiCache le risorse Amazon che richiedono AWS ServiceRoleForElastiCache

- Per eliminare un cluster, consultare i seguenti argomenti:
  - [Utilizzando il AWS Management Console](#)
  - [Utilizzo di AWS CLI per eliminare un ElastiCache cluster](#)
  - [Utilizzo del ElastiCache API](#)
- 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 \(\) ElastiCache API](#)

Eliminazione di un ruolo collegato ai servizi (console) IAM

È possibile utilizzare la IAM console per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato ai servizi (console)

1. Accedi AWS Management Console e apri la IAM console all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nel riquadro di navigazione della IAM console, scegli Ruoli. Quindi, seleziona la casella di controllo accanto al nome del ruolo che desideri eliminare, non il nome o la riga stessa.
3. In operazioni Role (Ruolo) nella parte superiore della pagina, seleziona Delete (Elimina) ruolo.
4. Nella finestra di dialogo di conferma, esamina i dati dell'ultimo accesso al servizio, che mostrano l'ultima volta che ciascuno dei ruoli selezionati ha effettuato l'ultimo accesso a un AWS servizio. In questo modo potrai verificare se il ruolo è attualmente attivo. Se desideri procedere, seleziona Yes, Delete (Sì, elimina) per richiedere l'eliminazione del ruolo collegato ai servizi.
5. Guarda le notifiche della IAM console per monitorare lo stato di avanzamento dell'eliminazione del ruolo collegato al servizio. Poiché l'eliminazione del ruolo IAM collegato al servizio è

asincrona, dopo aver inviato il ruolo per l'eliminazione, l'operazione di eliminazione può avere esito positivo o negativo. 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 al servizio () IAM CLI

È possibile utilizzare IAM le operazioni di AWS Command Line Interface per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato al servizio () 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 Names (ARNs) nel tuo account.

```
$ aws iam get-role --role-name role-name
```

Usa il nome del ruolo, non ilARN, per fare riferimento ai ruoli con le CLI operazioni. Ad esempio, se un ruolo ha il ARN `arn:aws:iam::123456789012:role/myrole`, si fa riferimento al ruolo come **myrole**.

2. Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `deletion-task-id` dalla risposta per controllare lo stato del task di eliminazione. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, inserire quanto segue:

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Inserire quanto segue per verificare lo stato del processo di eliminazione:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Lo stato di un task di eliminazione può essere NOT\_STARTED, IN\_PROGRESS, SUCCEEDED o FAILED. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema.

## Eliminazione di un ruolo collegato al servizio () IAM API

È possibile utilizzare il IAM API per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato al servizio () API

1. Per inviare una richiesta di cancellazione per un ruolo collegato a un servizio, chiama [DeleteServiceLinkedRole](#). Nella richiesta, specificare 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. Per verificare lo stato dell'eliminazione, chiama [GetServiceLinkedRoleDeletionStatus](#). 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.

## ElastiCache API autorizzazioni: riferimento ad azioni, risorse e condizioni

Quando si configurano criteri di [controllo degli accessi](#) e si scrivono criteri di autorizzazione da allegare a un IAM criterio (basato sull'identità o basato sulle risorse), utilizza la tabella seguente come riferimento. La tabella elenca ogni ElastiCache API operazione Amazon e le azioni corrispondenti per le quali puoi concedere le autorizzazioni per eseguire l'azione. Puoi specificare le operazioni nel campo `Action` della policy e il valore di una risorsa nel campo `Resource` della policy. Se non diversamente indicato, la risorsa è obbligatoria. Alcuni campi includono sia una risorsa obbligatoria che risorse facoltative. Quando non è presente alcuna risorsa ARN, la risorsa indicata nella policy è un carattere jolly (\*).

È possibile utilizzare i tasti di condizione nelle ElastiCache politiche per esprimere le condizioni. Per visualizzare un elenco di chiavi ElastiCache condizionali specifiche, insieme alle azioni e ai tipi di risorse a cui si applicano, consulta [Utilizzo delle chiavi di condizione](#). Per un elenco completo delle chiavi AWS-wide, consulta le chiavi di [contesto della condizione AWS globale nella Guida](#) per l'IAM utente.

### Note

Per specificare un'azione, utilizzate il `elasticache:` prefisso seguito dal nome dell'API operazione (ad esempio, `elasticache:DescribeCacheClusters`).

Per visualizzare un elenco di ElastiCache azioni, consulta [Actions Defined by Amazon ElastiCache](#) nel Service Authorization Reference.

## Convalida della conformità per Amazon ElastiCache

I revisori di terze parti valutano la sicurezza e la conformità dei AWS servizi nell'ambito di più programmi di AWS conformità SOC, come PCIRAMP, Fed e HIPAA.

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione [Ambito per programma di conformità Servizi AWS](#) di conformità e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta [Programmi di AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .



La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide introduttive su sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla AWS sicurezza e la conformità.
- [Architettura per la HIPAA sicurezza e la conformità su Amazon Web Services](#): questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee. HIPAA

#### Note

Non tutte sono idonee. Servizi AWS HIPAA Per ulteriori informazioni, consulta la [Guida ai servizi HIPAA idonei](#).

- [AWS Risorse per AWS](#) per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe riguardare il tuo settore e la tua località.
- [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
- [Evaluating Resources with Rules](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub](#)— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. 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](#).
- [Amazon GuardDuty](#): Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, ad esempio PCI DSS soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- [AWS Audit Manager](#)— Ciò Servizio AWS consente di verificare continuamente AWS l'utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

## Ulteriori informazioni

Per informazioni generali sulla conformità al AWS cloud, consulta quanto segue:

- [FIPSEndpoint per servizio](#)
- [Aggiornamenti del servizio in ElastiCache](#)
- [AWS Conformità al cloud](#)
- [Modello di responsabilità condivisa](#)
- [AWS PCIDSSProgramma di conformità](#)

## Resilienza in Amazon ElastiCache

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, è possibile progettare e gestire 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 tradizionali a data center singolo o multiplo.

[Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS](#)

Oltre all'infrastruttura AWS globale, Amazon ElastiCache offre diverse funzionalità per aiutarti a supportare le tue esigenze di resilienza e backup dei dati.

### 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 Memcached in esecuzione](#)
- [Mitigazione degli errori durante l'esecuzione di Valkey o Redis OSS](#)

- [Raccomandazioni](#)

## Limitazione degli errori con Memcached in esecuzione

Per minimizzare l'impatto degli errori con un motore Memcached in esecuzione, è possibile avvalersi delle seguenti soluzioni. Esistono due tipi di errori gestibili nei piani di mitigazione, gli errori dei nodi e quelli delle zone di disponibilità.

### Limitazione degli errori dei nodi

Le cache serverless mitigano automaticamente gli errori dei nodi con un'architettura multi-AZ replicata in modo che gli errori dei nodi siano trasparenti per l'applicazione. Per minimizzare l'impatto dell'errore di un nodo in un cluster progettato autonomamente, occorre innanzitutto distribuire i dati memorizzati nella cache tra più nodi. Poiché i cluster progettati autonomamente non supportano la replica, un errore di nodo comporta in ogni caso una perdita di dati del cluster.

Quando crei un cluster Memcached, puoi crearlo con da 1 a 60 nodi o più su richiesta speciale. Ripartire i dati tra più nodi consente di arginare la perdita di dati se un nodo genera un errore. Se, ad esempio, i dati vengono distribuiti tra dieci nodi, ciascuno di questi includerà circa il 10% dei dati memorizzati nella cache. Pertanto, in caso di errore, la perdita da arginare con la creazione e il provisioning di un nodo sostitutivo sarà del 10%. Se gli stessi dati fossero memorizzati in 3 nodi più grandi, un errore di nodo comporterebbe la perdita di circa il 33% dei dati memorizzati nella cache.

Per informazioni su come definire il numero di nodi in un cluster Memcached, consulta [Creazione di un cluster Memcached \(console\)](#).

### 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 nel maggior numero possibile di zone. Nell'improbabile eventualità di un errore AZ, perderai i dati memorizzati nella cache di quella AZ, non i dati memorizzati nella cache dell'altra. AZs

### Perché impostare tanti nodi?

Se la mia regione ha solo tre zone di disponibilità, perché mi occorrono più di tre nodi, dal momento che, se una zona genera un errore, perdo comunque un terzo dei dati?

Ottima domanda. Esistono due tipi di errori di cui limitare gli effetti, gli errori dei nodi e quelli delle zone di disponibilità. Effettivamente, se i dati sono distribuiti su più zone di disponibilità, una delle quali genera un errore, si perdono solo i dati memorizzati in quest'ultima, indipendentemente dal numero di nodi in essere. Tuttavia, se è un nodo a generare l'errore, la presenza di più nodi riduce proporzionalmente la perdita di dati.

Non esiste una "formula magica" che consenta di determinare il numero di nodi da configurare in un cluster. È necessario valutare l'eventuale impatto della perdita di dati rispetto alla probabilità di errore e ai costi, per poi trarre le proprie conclusioni.

Per informazioni su come definire il numero di nodi in un cluster Memcached, consulta [Creazione di un cluster Memcached \(console\)](#).

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

## Mitigazione degli errori durante l'esecuzione di Valkey o Redis OSS

Quando si utilizza un OSS motore Valkey o Redis, sono disponibili le seguenti opzioni per ridurre al minimo l'impatto di un errore di un nodo o di una zona di disponibilità.

### 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 mitigare l'impatto dei guasti dei OSS nodi Valkey o Redis sui cluster progettati autonomamente, sono disponibili le seguenti opzioni:

### Argomenti

- [Attenuazione degli errori: Valkey o Redis Replication Groups OSS](#)

### Attenuazione degli errori: Valkey o Redis Replication Groups OSS

Un gruppo di OSS replica Valkey o Redis è composto da un singolo nodo primario dal quale l'applicazione può sia leggere che scrivere, e da 1 a 5 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.

## Valkey o Redis Multi-AZ OSS

È possibile abilitare Multi-AZ sui gruppi di replica Valkey o Redis. OSS 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 al minimo dei tempi di inattività ElastiCache utilizzando Multi-AZ con Valkey e Redis OSS](#).

### 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 una maggiore protezione, si consiglia di avviare i nodi del gruppo di replica in diverse zone di disponibilità (). AZs 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. Tuttavia, se si posizionano i nodi in più aree di AZs disponibilità, in caso di errore in una zona di disponibilità si perderanno solo i nodi di tale zona.

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 su come specificare le zone di disponibilità per i OSS nodi Valkey o Redis, consulta [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#)

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

## 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 di un errore di nodo quando si utilizza Valkey o RedisOSS, si consiglia di utilizzare nell'implementazione più nodi in ogni shard e di distribuire i nodi su più zone di disponibilità. Questa operazione viene eseguita automaticamente per le cache serverless.

Per i cluster progettati autonomamente su Valkey o RedisOSS, consigliamo di abilitare Multi-AZ sul gruppo di replica in modo da eseguire automaticamente il failover su una replica in caso di guasto del ElastiCache nodo primario.

Quando si esegue Memcached e si ripartiscono i dati tra i nodi, più nodi si configurano, minore sarà la perdita di dati in caso di errore.

## 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. La distribuzione uniforme dei nodi tra i nodi AZs ridurrà al minimo l'impatto nell'improbabile caso di un errore AZ. Questa operazione viene eseguita automaticamente per le cache serverless.

## Altre precauzioni

Se utilizzi Valkey o RedisOSS, oltre a quanto sopra, ti consigliamo di pianificare 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

In quanto servizio gestito, AWS ElastiCache è protetto dalle procedure di sicurezza di rete AWS globali descritte nella sezione Sicurezza e conformità di [AWS Architecture Center](#).

Si utilizzano API chiamate AWS pubblicate per accedere ElastiCache tramite la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versione successiva. Consigliamo TLS 1.3 o versioni successive. I client devono inoltre supportare suite di crittografia con Perfect Forward Secrecy (PFS) come Ephemeral Diffie-Hellman () o Elliptic Curve Ephemeral Diffie-Hellman (). DHE 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.

### Argomenti

- [Gestione degli aggiornamenti del servizio per cluster progettati autonomamente](#)

## Gestione degli aggiornamenti del servizio per cluster progettati autonomamente

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 vengono rilasciati gli aggiornamenti. 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:



## 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 available o scheduled.

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 o AWS CLI applicare API 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 a AWS Management Console e apri la ElastiCache console Amazon all'indirizzo <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 i tuoi cluster ElastiCache (RedisOSS) stiano eseguendo l'ultimo aggiornamento del servizio seguendo questi passaggi:

1. Scegli un cluster applicabile nella pagina `Redis OSS 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-b...	Status	Cluster ...
elasticache-redis-6-2-6-update-202310109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 03:59:59 (UT...	Available	March 1, 2...

Se sulla console viene visualizzato il messaggio «Nessun aggiornamento del servizio trovato», significa che al cluster ElastiCache (RedisOSS) è già applicato l'ultimo aggiornamento del servizio.

Service update name	Cluster...	Update type	Update severity	Release date	Recommended ...
No service updates found.					

## 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 ElastiCache

Common Vulnerabilities and Exposures (CVE) è un elenco di voci relative a vulnerabilità di sicurezza informatica note pubblicamente. Ogni voce è un link che contiene un numero di identificazione, una descrizione e almeno un riferimento pubblico. In questa pagina è disponibile un elenco delle vulnerabilità di sicurezza che sono state risolte in ElastiCache.

Ti consigliamo di eseguire sempre l'aggiornamento alle versioni più recenti di ElastiCache Valkey, Redis OSS o ElastiCache Memcached per proteggerti dalle vulnerabilità note. Quando si utilizza una cache ElastiCache Serverless, le CVE correzioni vengono applicate automaticamente alla cache. Quando si utilizzano cluster progettati autonomamente con Valkey o RedisOSS, espone il componente. ElastiCache PATCH Ad esempio, quando si utilizza la versione 6.2.6 ElastiCache (RedisOSS), la versione principale è 6, la versione secondaria è 2 e la versione patch è 6. PATCHLe versioni riguardano correzioni di bug, correzioni di sicurezza e modifiche non funzionali compatibili con le versioni precedenti.

È possibile utilizzare la tabella seguente per verificare se una particolare versione di ElastiCache Valkey e Redis OSS ha una correzione per una specifica vulnerabilità di sicurezza. Se il tuo OSS cluster ElastiCache Valkey o Redis esegue una versione senza la correzione di sicurezza, fai riferimento alla tabella seguente e agisci. Puoi eseguire l'aggiornamento a una OSS versione ElastiCache Valkey o Redis più recente contenente la correzione oppure, se utilizzi una versione contenente la correzione, assicurati di aver applicato l'ultimo aggiornamento del servizio facendo riferimento a [Gestione degli aggiornamenti del servizio per cluster progettati autonomamente](#). Per ulteriori informazioni sulle versioni del ElastiCache motore supportate e su come eseguire l'aggiornamento, consulta [Versioni del motore e aggiornamento in ElastiCache](#)

## Note

- Se a CVE è indirizzato in una ElastiCache versione, significa che è indirizzato anche nelle versioni più recenti. Ad esempio, se una vulnerabilità viene risolta nella versione 6.0.5 ElastiCache (RedisOSS), questa situazione continua nelle versioni 6.2.6, 7.0.7 e 7.1.
- Un asterisco (\*) nella tabella seguente indica che è necessario applicare l'ultimo aggiornamento del servizio per il cluster (Redis) che esegue la versione ElastiCache (RedisOSS) specificata per risolvere la ElastiCache vulnerabilità di OSS sicurezza. Per ulteriori informazioni su come verificare che sia stato applicato l'ultimo aggiornamento del

servizio per la versione ElastiCache (RedisOSS) su cui è in esecuzione il cluster, consulta.  
[Gestione degli aggiornamenti del servizio per cluster progettati autonomamente](#)

ElastiCache versione (RedisOSS)	CVEsIndirizzato
Redis 6.0.5 OSS	<a href="#">CVE-2022-24735</a> *, <a href="#">-2022-24736</a> * <a href="#">CVE</a>
Redis OSS 6.2.6	<a href="#">CVE-2022-24834</a> *, <a href="#">-2022-35977</a> *, <a href="#">-2022-36021</a> *, <a href="#">CVE-2022-24735</a> , <a href="#">-2022-24736</a> <a href="#">CVE</a> <a href="#">CVE</a> <a href="#">CVE</a>
OSSRedis 7.0.7	<a href="#">CVE-2023-41056</a> *, <a href="#">-2022-24834</a> *, <a href="#">-2022-35977</a> , <a href="#">CVE-2022-36021</a> , <a href="#">-2022-24735</a> , <a href="#">-2022-24736</a> <a href="#">CVE</a> <a href="#">CVE</a> <a href="#">CVE</a> <a href="#">CVE</a>
OSSRedis 7.1.0	<a href="#">CVE-2023-41056</a> , <a href="#">-2022-24834</a> , <a href="#">-2022-35977</a> , <a href="#">CVE-2022-36021</a> , <a href="#">-2022-24735</a> , <a href="#">-2022-24736</a> <a href="#">CVE</a> <a href="#">CVE</a> <a href="#">CVE</a> <a href="#">CVE</a>

# Registrazione e monitoraggio in Amazon ElastiCache

Ai fini della gestione è importante conoscere le prestazioni delle cache. ElastiCache genera metriche pubblicate su Amazon CloudWatch Logs per monitorare le prestazioni della cache. Inoltre, ElastiCache genera eventi quando si verificano cambiamenti significativi nelle risorse della cache (ad esempio, viene creata una nuova cache o viene eliminata una cache).

## Argomenti

- [Metriche ed eventi serverless per Valkey e Redis OSS](#)
- [Metriche ed eventi serverless per Memcached](#)
- [Registrazione delle ElastiCache API chiamate Amazon con AWS CloudTrail](#)
- [SNSMonitoraggio degli ElastiCache eventi da Amazon](#)
- [Consegna dei registri](#)
- [Monitoraggio dell'utilizzo con CloudWatch Metrics](#)
  
- [Registrazione delle ElastiCache API chiamate Amazon con AWS CloudTrail](#)

## Metriche ed eventi serverless per Valkey e Redis OSS

ElastiCache offre un'ampia varietà di metriche ed eventi per il monitoraggio quando si lavora con cache serverless. Ciò include CloudWatch metriche, metriche a livello di comando e registri degli eventi che possono essere acquisiti tramite Amazon EventBridge

## Argomenti

- [Metriche delle cache serverless](#)
- [Eventi di cache serverless](#)
- [Metriche ed eventi del cluster progettati autonomamente per Valkey e Redis OSS](#)

## Metriche delle cache serverless

Il AWS/ElastiCache namespace include le seguenti CloudWatch metriche per le cache serverless Valkey o Redis. OSS

Codici metrici per Valkey o Redis OSS



Parametro	Descrizione	Unità
BytesUsedForCache	Numero totale di byte utilizzati dai dati archiviati nella cache.	Byte
ElastiCacheProcessingUnits	Il numero totale di ElastiCacheProcessingUnits (ECPUs) consumati dalle richieste eseguite nella cache	Conteggio
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	Il numero di richieste che sono state limitate ElastiCache perché il carico di lavoro si stava scalando più velocemente	Conteggio

Parametro	Descrizione	Unità
	te di quanto possa scalare. ElastiCache	
NewConnections	Il numero totale di connessioni accettate dal server durante questo periodo.	Conteggio
Currltems	Il numero di elementi nella cache.	Conteggio
CurrVolatileltems	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 espulse dalla cache.	Conteggio
IamAuthenticationExpirations	Il numero totale di connessioni Valkey o IAM Redis autenticazione scadute. OSS Ulteriori informazioni su <a href="#">Autenticazione con IAM</a> sono disponibili nella guida per l'utente.	Conteggio
IamAuthenticationThrottling	Il numero totale di richieste o richieste Valkey o Redis con autenticazione limitata IAM. OSS AUTH HELLO Ulteriori informazioni su <a href="#">Autenticazione con IAM</a> 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	Il numero totale di tentativi falliti di autenticazione su Valkey o Redis utilizzando il comando. OSS 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 emette anche le seguenti metriche a livello di comando. Per ogni tipo di comando, ElastiCache emette il numero totale di comandi e il numero di comandi ECPUs utilizzati da quel tipo di comando.

Parametro	Descrizione	Unità
EvalBasedCmds	Il numero di comandi get che la cache ha ricevuto.	Conteggio
EvalBasedCmdsECPUs	ECPUsconsumato dai comandi basati su eval.	Conteggio
GeoSpatialBasedCmds	Numero totale di comandi per i comandi basati su GeoSpace. Questo è derivato dalla statistica commandstats di Valkey o RedisOSS. sommando tutti i tipi di comandi geo: geoadd, geodist, geohash, geopos, georadius e georadiusbymember.	Conteggio
GeoSpatialBasedCmdsECPUs	ECPUsconsumato da comandi basati su dati geospaziali.	Conteggio
GetTypeCmds	Il numero totale di comandi di tipo sola lettura. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi di tipo di sola lettura (get, hget, scard, lrange e così via).	Conteggio
GetTypeCmdsECPUs	ECPUsconsumato dai comandi di lettura.	Conteggio
HashBasedCmds	Il numero totale di comandi basati su hash. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i	Conteggio

Parametro	Descrizione	Unità
	comandi che agiscono su uno o più hash (hget, hkeys, hvals, hdel e così via).	
HashBasedCmdsECPUs	ECPUs consumato da comandi basati su hash.	Conteggio
HyperLogLogBasedCmds	Il numero totale di comandi HyperLogLog basati. Viene derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi di tipo pf (pfadd, pfcount, pfmerge e così via).	Conteggio
HyperLogLogBasedCmdsECPUs	ECPUs consumato dai comandi basati. HyperLogLog	Conteggio
JsonBasedCmds	Il numero totale di JSON comandi, inclusi i comandi di lettura e scrittura. È derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i JSON comandi che agiscono sui tasti. JSON	Conteggio
JsonBasedCmdsECPUs	ECPUs utilizzato da tutti i JSON comandi, inclusi i comandi di lettura e scrittura.	Conteggio

Parametro	Descrizione	Unità
JsonBasedGetCmds	Il numero totale di comandi di JSON sola lettura. È derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi di lettura che agiscono sulle chiavi. JSON JSON	Conteggio
JsonBasedGetCmdsECPUs	ECPUsJSONconsumato da comandi di sola lettura.	Conteggio
JsonBasedSetCmds	Il numero totale di comandi di JSON scrittura. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi di JSON scrittura che agiscono sulle chiavi. JSON	Conteggio
JsonBasedSetCmdsECPUs	ECPUsJSONconsumato dai comandi di scrittura.	Conteggio
KeyBasedCmds	Il numero totale di comandi basati su chiavi. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi che agiscono su una o più chiavi su più strutture di dati (del, expire, rename e così via).	Conteggio
KeyBasedCmdsECPUs	ECPUsconsumato da comandi basati su chiavi.	Conteggio

Parametro	Descrizione	Unità
ListBasedCmds	Il numero totale di comandi basati su elenchi. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi che agiscono su uno o più elenchi (lindex, lrange, lpush, ltrim e così via).	Conteggio
ListBasedCmdsECPUs	ECPUsconsumato dai comandi basati su elenchi.	Conteggio
NonKeyTypeCmds	Il numero totale di comandi non basati su chiavi. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi che non agiscono su una chiave, ad esempio acl, dbsize o info.	Conteggio
NonKeyTypeCmdsECPUs	ECPUsconsumato dai comandi. non-key-based	Conteggio
PubSubBasedCmds	Numero totale di comandi per la funzionalità pub/sub. Questo è derivato dal OSS commandstatsstatistics di Valkey o Redis sommando tutti i comandi utilizzati per la funzionalità pub/sub: psubscribe, publish, pubsub, punsubscribe, ssubscribe e, sunsubscribe, spublish, subscribe e unsubscribe.	Conteggio

Parametro	Descrizione	Unità
PubSubBasedCmdsECPUs	ECPUsconsumato dai comandi basati su pub/sub.	Conteggio
SetBasedCmds	Il numero totale di comandi basati su set. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi che agiscono su uno o più set (scard, sdiff, sadd, sunion e così via).	Conteggio
SetBasedCmdsECPUs	ECPUsconsumato dai comandi basati su set.	Conteggio
SetTypeCmds	Il numero totale di comandi di tipo scrittura. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i tipi mutativi di comandi che operano sui dati (set, hset, sadd, lpop e così via).	Conteggio
SetTypeCmdsECPUs	ECPUsconsumato dai comandi di scrittura.	Conteggio
SortedSetBasedCmds	Il numero totale di comandi basati su set ordinati. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi che agiscono su uno o più set ordinati (zcount, zrange, zrank, zadd e così via).	Conteggio



Parametro	Descrizione	Unità
SortedSetBasedCmdsECPUs	ECPUsconsumato da comandi basati su ordinamenti.	Conteggio
StringBasedCmds	Il numero totale di comandi basati su stringhe. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi che agiscono su una o più stringhe (strlen, setex, setrange e così via).	Conteggio
StringBasedCmdsECPUs	ECPUsconsumato da comandi basati su stringhe.	Conteggio
StreamBasedCmds	Il numero totale di comandi basati sul flusso. Questo è derivato dalla statistica OSS commandstats di Valkey o Redis sommando tutti i comandi che agiscono su uno o più tipi di dati di stream (xrange, xlen, xadd, xdel e così via).	Conteggio
StreamBasedCmdsECPUs	ECPUsconsumato da comandi basati su stream.	Conteggio

## Eventi di cache serverless

ElastiCache registra gli eventi relativi alla cache serverless. Queste informazioni includono la data, l'ora, il nome e tipo di fonte e una descrizione dell'evento. È possibile recuperare facilmente gli eventi dal registro utilizzando la ElastiCache console, il comando AWS CLI `describe-events` o l'azione.

ElastiCache API `DescribeEvents`

Puoi scegliere di monitorare, importare, trasformare e intervenire sugli ElastiCache eventi utilizzando Amazon EventBridge. Scopri di più nella pagina Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

## Visualizzazione degli eventi (Console) ElastiCache

Per visualizzare gli eventi utilizzando la ElastiCache console:

1. Accedi AWS Management Console e apri la ElastiCache console 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 evento e visualizza l'origine dell'evento, il tipo di evento, l'GMTora dell'evento e una 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 ElastiCache degli eventi (AWS CLI)

Per generare un elenco di ElastiCache eventi utilizzando 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 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>. Indirizzi IP gratuiti 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>. Limite di quota

Tipo di evento	Descrizione	Unità	Origine	Messaggio
				raggiunto per la creazione dell'VPCe ndpoint.
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Non disponi delle autorizzazioni per creare un VPC endpoint.
Creazione della cache non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Un utente con una OSS versione Valkey o Redis incompatibile è presente nel gruppo di utenti < >. user-group-name
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 < user-group-name > 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 &lt;nome-cache&gt;. Il ripristino dei dati dallo snapshot non è riuscito perché &lt;motivo&gt;.</p> <p>Motivi dell'errore:</p> <ul style="list-style-type: none"> <li>recupero del file da S3 non riuscito;</li> <li>l'hash md5 previsto non corrisponde a quello effettivo ;</li> <li>il RDB file fornito ha una versione non supportata.</li> </ul>

### Eventi di aggiornamento della cache senza server (Valkey o Redis) OSS

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Aggiornamento della cache completato	ARN della cache	modifica della configurazione	Cache serverless	SecurityGroups <cache-name>aggiornato per la 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 OSS versione Valkey o Redis incompatibile è presente nel gruppo di utenti <>. user-group-name
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	<cache-name>Un aggiornamento della cache non è riuscito. SecurityGroups aggiornamento non riuscito.

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	Un aggiornamento della cache <cache-name> non è riuscito. SecurityGroups aggiornamento non 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. SecurityGroups aggiornamento non riuscito perché non SecurityGroups sono validi.

### Eventi di eliminazione della cache senza server (Valkey o Redis) OSS

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 della cache serverless (Valkey o Redis) OSS

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 serverless Cache Snapshot (Valkey o Redis) OSS

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"> <li>• perché la chiave gestita dal cliente è disattivata;</li> <li>• perché la chiave gestita dal cliente non è stata trovata;</li> <li>• perché si è verificato il timeout della richiesta.</li> </ul>
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 &lt;nome-cache&gt;. La creazione dello snapshot &lt;nome-snapshot&gt; non è riuscita perché &lt;motivo&gt;.</p> <p>Motivo predefinito:</p> <ul style="list-style-type: none"> <li>• a causa di un errore interno.</li> </ul>

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 l'istantanea 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 del cluster progettati autonomamente per Valkey e Redis OSS

ElastiCache offre un'ampia varietà di metriche ed eventi per il monitoraggio di cluster progettati autonomamente quando si lavora con Valkey e Redis. OSS Ciò include metriche a livello di host, metriche a livello di comando e registri degli eventi disponibili tramite Amazon AWS CLI Simple Notification Service (). SNS

## Argomenti

- [Metriche per cluster progettati autonomamente](#)
- [Eventi per cluster progettati autonomamente \(Valkey e Redis\) OSS](#)

## Metriche per cluster progettati autonomamente

Quando progetti autonomamente i cluster, ElastiCache emette metriche a ogni livello di nodo, incluse le metriche a livello di host e le metriche 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 [Metriche per Valkey e Redis OSS](#).

## Eventi per cluster progettati autonomamente (Valkey e Redis) OSS

ElastiCache registra gli eventi che si riferiscono alle cache progettate autonomamente. Quando lavori con cluster progettati autonomamente, puoi visualizzare gli eventi del cluster nella ElastiCache console, utilizzando o utilizzando Amazon Simple AWS CLI Notification Service ()SNS. Gli eventi 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. Puoi recuperare facilmente gli eventi dal registro utilizzando la ElastiCache console, il comando AWS CLI describe-events o l'azione. ElastiCache API DescribeEvents

## Visualizzazione degli eventi (Console) ElastiCache

La procedura seguente visualizza gli eventi utilizzando la ElastiCache console.

Per visualizzare gli eventi utilizzando la ElastiCache console



1. Accedi AWS Management Console e apri la ElastiCache console 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 evento e visualizza l'origine dell'evento, il tipo di evento, l'GMTora dell'evento e una 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 ElastiCache degli eventi (AWS CLI)

Per generare un elenco di ElastiCache eventi utilizzando 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 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 ElastiCache eventi attivano SNS le notifiche Amazon. Per ulteriori informazioni sui dettagli degli eventi, consulta [Visualizzazione ElastiCache degli eventi](#).

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 a causa di indirizzi IP gratuiti insufficienti	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 a causa di uno stato di rete incompatibile	ElastiCache: CacheClusterProvisioningFailed : <i>cluster-name</i>	È stato effettuato un tentativo di avviare un nuovo cluster di cache in un cloud privato virtuale inesistente (). VPC
ElastiCache: CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Dimensionamento di un cluster di cache completato.
ElastiCache: CacheClusterScalingFailed	ElastiCache: CacheClusterScalingFailed : <i>cluster-name</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"><li>• L'elenco dei gruppi di sicurezza cache autorizzati per il cluster di cache è stato modificato.</li><li>• Uno o più nuovi gruppi EC2 di sicurezza sono stati autorizzati su uno qualsiasi dei gruppi di sicurezza della cache associati al cluster di cache.</li><li>• Uno o più gruppi EC2 di sicurezza sono stati revocati da uno qualsiasi dei gruppi di sicurezza della cache associati al cluster di cache.</li></ul>

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 e ha iniziato a sostituire il nodo di cache.</p> <div data-bbox="1068 495 1507 810"><p> <b>Note</b></p><p>La DNS voce relativa al nodo di cache sostituito non viene modificata.</p></div> <p>Nella maggior parte dei casi, non è necessario aggiornar e l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client di cache potrebbero smettere di utilizzare il nodo di cache anche dopo ElastiCache aver sostituito il nodo di cache; in questo caso, l'applicazione dovrebbe aggiornare l'elenco dei server quando si verifica questo evento.</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 e ha completato la sostituzione del nodo di cache.</p> <div data-bbox="1068 541 1507 856"><p> <b>Note</b></p><p>La DNS voce relativa al nodo di cache sostituito non viene modificata.</p></div> <p>Nella maggior parte dei casi, non è necessario aggiornare e l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client di cache potrebbero smettere di utilizzare il nodo di cache anche dopo ElastiCache aver sostituito il nodo di cache; in questo caso, l'applicazione dovrebbe aggiornare l'elenco dei server quando si verifica questo evento.</p>

Nome evento	Messaggio	Descrizione
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Uno o più nodi di cache sono stati riavviati.  Messaggio (Memcached): "Cache node %s shutdown" Secondo messaggio: "Cache node %s restarted"
ElastiCache: CertificateRenewalComplete (solo Valkey o Redis) OSS	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 Valkey o Redis) OSS	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.

Nome evento	Messaggio	Descrizione
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	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.
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 <a href="#">Sostituzione dei nodi (Valkey e RedisOSS)</a> .
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 <a href="#">Sostituzione dei nodi (Valkey e RedisOSS)</a> .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Un nodo di cache è stato rimosso dal cluster di cache.

Nome evento	Messaggio	Descrizione
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.
ElastiCache: SnapshotComplete (solo Valkey o Redis) OSS	ElastiCache:SnapshotComplete : <i>cluster-name</i>	È stata correttamente acquisita una snapshot di cache.
ElastiCache: SnapshotFailed (solo Valkey o Redis) OSS	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 <a href="#">DescribeSnapshots</a>, sarà failed.</p>



# Metriche ed eventi serverless per Memcached

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

Questa sezione descrive le metriche e gli eventi che è possibile monitorare quando si lavora con le cache serverless Memcached.

Lo spazio dei nomi AWS/ElastiCache include le seguenti CloudWatch metriche per le cache serverless Memcached.

Parametro	Descrizione	Unità
BytesUsedForCache	Numero totale di byte utilizzati dai dati archiviati nella cache.	Byte
ElastiCacheProcessingUnits	Il numero totale di ElastiCacheProcessingUnits (ECPUs) consumati dalle richieste eseguite nella cache	Conteggio
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
CurrConnections	Numero di connessioni client alla cache.	Conteggio

Parametro	Descrizione	Unità
ThrottledCmds	Il numero di richieste che sono state limitate ElastiCache perché il carico di lavoro si stava scalando più velocemente di quanto possa scalare. ElastiCache	Conteggio
NewConnections	Il numero totale di connessioni accettate dal server durante questo periodo.	Conteggio
Currltems	Il numero di elementi nella cache.	Conteggio
NetworkBytesIn	Byte totali trasferiti nella cache.	Byte
NetworkBytesOut	Byte totali trasferiti dalla cache.	Byte
Espulsioni	Numero di chiavi espulse dalla cache.	Conteggio
Reclaimed	Numero di chiavi scadute della cache.	Conteggio

### Metriche a livello di comando

ElastiCache emette anche le seguenti metriche a livello di comando Memcached

Parametro	Descrizione	Unità
CmdGet	Il numero di comandi get che la cache ha ricevuto.	Conteggio

Parametro	Descrizione	Unità
CmdSet	Il numero di comandi stabiliti che la cache ha ricevuto.	Conteggio
CmdTouch	Il numero di comandi touch ricevuti dalla cache.	Conteggio
GetHits	Il numero di richieste di recupero che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio
GetMisses	Il numero di richieste di recupero che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
IncrHits	Il numero di richieste di incremento che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio
IncrMisses	Il numero di richieste di incremento che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
DecrHits	Il numero di richieste di decremento che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio
DecrMisses	Il numero di richieste di decremento che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio

Parametro	Descrizione	Unità
DeleteHits	Il numero di richieste di eliminazione che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio
DeleteMisses	Il numero di richieste di eliminazione che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
TouchHits	Il numero di chiavi che sono state toccate e a cui è stata assegnata una nuova ora di scadenza.	Conteggio
TouchMisses	Il numero di chiavi toccate, ma non trovate.	Conteggio
CasHits	Il numero di richieste Cas ricevute dalla cache dove la chiave richiesta è stata trovata e il valore Cas corrispondeva.	Conteggio
CasMisses	Il numero di richieste Cas ricevute dalla cache dove la chiave richiesta non è stata trovata.	Conteggio
CasBadval	Il numero di richieste Cas ricevute dalla cache dove il valore Cas non corrispondeva al valore Cas archiviato.	Conteggio
CmdFlush	Il numero di comandi flush che la cache ha ricevuto.	Conteggio

## Eventi di cache serverless

ElastiCache registra gli eventi relativi alla cache serverless. Queste informazioni includono la data, l'ora, il nome e tipo di fonte e una descrizione dell'evento. È possibile recuperare facilmente gli eventi dal registro utilizzando la ElastiCache console, il comando AWS CLI `describe-events` o l'azione ElastiCache API `DescribeEvents`.

ElastiCache API `DescribeEvents`

Puoi scegliere di monitorare, importare, trasformare e intervenire sugli ElastiCache eventi utilizzando Amazon EventBridge. Scopri di più nella pagina Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

### Visualizzazione degli eventi (Console) ElastiCache

Per visualizzare gli eventi utilizzando la ElastiCache console:

1. Accedi AWS Management Console e apri la ElastiCache console 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 evento e visualizza l'origine dell'evento, il tipo di evento, l'GMTora dell'evento e una 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 ElastiCache degli eventi (AWS CLI)

Per generare un elenco di ElastiCache eventi utilizzando 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 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 non riuscita	ARN della cache	errore	Cache serverless	Impossibile creare la cache <nome-cache>. Indirizzi IP gratuiti 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>. Limite di quota raggiunto per la creazione dell'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>. Non disponi delle autorizzazioni per creare un VPC endpoint.

### Eventi di aggiornamento della cache senza server (Memcached)

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Aggiornamento della cache completato	ARN della cache	modifica della configurazione	Cache serverless	SecurityGroups <cache-name>aggiornato per la cache.
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	Un aggiornamento della cache <cache-name>non è riuscito. SecurityGroups aggiornamento non riuscito.
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	Un aggiornamento della cache <cache-name>non

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
				è riuscito. SecurityGroups aggiornamento non riuscito a causa di autorizzazioni insufficienti.
Aggiornamento della cache non riuscito	ARN della cache	modifica della configurazione	Cache serverless	Un aggiornamento della cache <cache-name> non è riuscito. SecurityGroups aggiornamento non riuscito perché non SecurityGroups sono validi.

#### Eventi di eliminazione della cache senza server (Memcached)

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 della cache serverless (Memcached)



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>.
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.

### Eventi Serverless Cache Snapshot (Memcached)

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	errore	serverless-cache-snapshot	Impossibile creare lo snapshot

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
	ARN dello snapshot			<p>per la cache &lt;nome-cache&gt;. Creazione dello snapshot &lt;nome-snapshot&gt; non riuscita con la chiave gestita dal cliente &lt;id-chiave&gt; &lt;motivo&gt;.</p> <p>Messaggi del motivo dell'errore:</p> <ul style="list-style-type: none"><li>• perché la chiave gestita dal cliente è disattivata;</li><li>• perché la chiave gestita dal cliente non è stata trovata;</li><li>• perché si è verificato il timeout della richiesta.</li></ul>

Tipo di evento	Elenco delle risorse	Categoria	Origine	Messaggio
Creazione dello snapshot non riuscita	ARN della cache ARN dello snapshot	errore	serverless-cache-snapshot	<p>Impossibile creare lo snapshot per la cache &lt;nome-cache&gt;. La creazione dello snapshot &lt;nome-snapshot&gt; non è riuscita perché &lt;motivo&gt;.</p> <p>Motivo predefinito:</p> <ul style="list-style-type: none"> <li>• a causa di un errore interno.</li> </ul>
Esportazione dello snapshot non riuscita	ARN dello snapshot	errore	serverless-cache-snapshot	<p>Impossibile esportare lo snapshot per la cache &lt;nome-cache&gt;. Impossibile esportare l'istantanea nel bucket %s perché ElastiCache non dispone delle autorizzazioni per il bucket.</p>

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 nel bucket è già presente un oggetto con lo stesso nome.
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.

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 S3 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 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 esiste.
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>.

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".
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>.

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 destinazione "%s" "%s".

## Registrazione delle ElastiCache API chiamate Amazon con AWS CloudTrail

Amazon ElastiCache è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon ElastiCache. CloudTrail acquisisce tutte le API chiamate per Amazon ElastiCache come eventi, incluse le chiamate dalla ElastiCache console Amazon e le chiamate in codice alle ElastiCache API operazioni Amazon. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amazon ElastiCache. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon ElastiCache, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la [Guida AWS CloudTrail per l'utente](#).



## ElastiCache Informazioni su Amazon in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in Amazon ElastiCache, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon ElastiCache, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi da tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un trail](#)
- [CloudTrail Servizi e integrazioni supportati](#)
- [Configurazione di Amazon SNS Notifications per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

[Tutte le ElastiCache azioni di Amazon vengono registrate CloudTrail e documentate nel Reference. ElastiCache API](#) Ad esempio, le chiamate a `DescribeCacheCluster` e `CreateCacheCluster` le `ModifyCacheCluster` azioni generano voci nei file di CloudTrail registro.

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 credenziali root o IAM utente.
- 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 AWS servizio.

Per ulteriori informazioni, consulta l'[CloudTrail userIdentityelemento](#).

## Comprendere le voci dei file di ElastiCache log di Amazon

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle API chiamate pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'CreateCacheClusterazione.

```
{
 "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 CloudTrail registro che illustra l'DescribeCacheClusterazione. Tieni presente che per tutte le chiamate Amazon ElastiCache Descrivi (Describe\*), la ResponseElements sezione viene rimossa e appare comenu11.

```

{
 "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 CloudTrail registro che registra un'ModifyCacheClusterazione.

```

{
 "eventVersion":"1.01",
 "userIdentity":{
 "type":"IAMUser",
 "principalId":"EXAMPLEEXAMPLEEXAMPLE",
 "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
 "accountId":"123456789012",
 "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
 "userName":"elasticache-allow"
 },
 "eventTime":"2014-12-01T22:32:21Z",
 "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"
}
```

## SNSMonitoraggio degli ElastiCache eventi da Amazon

Quando si verificano eventi significativi per un cluster, ElastiCache invia una notifica a uno specifico SNS argomento di Amazon. Esempi includono l'impossibilità di aggiungere un nodo, l'aggiunta di un nodo, la modifica di un gruppo di sicurezza e altro ancora. Tramite il monitoraggio degli eventi chiave, è possibile conoscere lo stato corrente dei cluster e, in base all'evento, intraprendere eventuali operazioni correttive.

### Argomenti

- [Gestione delle SNS notifiche ElastiCache Amazon](#)
- [Visualizzazione ElastiCache degli eventi](#)

- [Notifiche di eventi e Amazon SNS](#)

## Gestione delle SNS notifiche ElastiCache Amazon

Puoi configurare l'invio ElastiCache di notifiche per importanti eventi del cluster utilizzando Amazon Simple Notification Service (AmazonSNS). In questi esempi, configurerai un cluster con l'Amazon Resource Name (ARN) di un SNS argomento Amazon per ricevere notifiche.

### Note

- Questo argomento presuppone che tu ti sia registrato ad Amazon SNS e che tu abbia impostato e sottoscritto un argomento AmazonSNS. Per ulteriori informazioni su come procedere, consultare la [Guida per gli sviluppatori di Amazon Simple Notification Service](#).
- Per impostazione predefinita, ha API `modify-replication-group` effetto su tutti i gruppi di una regione e non solo sul gruppo attualmente specificato. Se desideri configurare un gruppo specifico in una regione in modo diverso rispetto agli altri gruppi, puoi utilizzare l'opzione `--notification-topic-arn` per creare un argomento separato per quel gruppo.

## Aggiungere un SNS argomento Amazon

Le seguenti sezioni mostrano come aggiungere un SNS argomento Amazon utilizzando la AWS Console, il AWS CLI, o il ElastiCache API.

### Aggiungere un SNS argomento Amazon (Console)

La procedura seguente mostra come aggiungere un SNS argomento Amazon per un cluster. Quando usi Valkey o Redis per OSS aggiungere un SNS argomento Amazon per un gruppo di replica nella fase 2, invece di scegliere un cluster, scegli un gruppo di replica. Quindi segui gli stessi passaggi rimanenti.

### Note

Questo processo può essere utilizzato anche per modificare l'SNSargomento Amazon.

Per aggiungere o modificare un SNS argomento Amazon per un cluster (console)

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. In Clusters, scegli il cluster per il quale desideri aggiungere o modificare un SNS argomento ARN Amazon.
3. Scegli Modifica.
4. In Modifica cluster in Argomento per la SNS notifica, scegli l'SNSargomento che desideri aggiungere oppure scegli ARNInput manuale e digita ARN l'SNSargomento Amazon.
5. Scegli Modifica.

Aggiungere un SNS argomento Amazon (AWS CLI)

Per aggiungere o modificare un SNS argomento Amazon per un cluster, usa il AWS CLI comandomodify-cache-cluster.

Il seguente esempio di codice aggiunge un SNS argomento Amazon arn 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](#)

Aggiungere un SNS argomento Amazon (ElastiCache API)

Per aggiungere o modificare un SNS argomento Amazon per un cluster, richiama l'ModifyCacheClusterazione con i seguenti parametri:

- CacheClusterId=my-cluster

- TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications

## Example

```
https://elasticache.amazonaws.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](#).

## Abilitazione e disabilitazione delle notifiche Amazon SNS

È possibile, in base alle proprie esigenze, abilitare o disabilitare le notifiche relative a un cluster. Le seguenti procedure mostrano come disabilitare SNS le notifiche Amazon.

### Abilitazione e disabilitazione SNS delle notifiche Amazon (Console)

Per disabilitare SNS le notifiche di Amazon utilizzando il AWS Management Console

1. Accedi a AWS Management Console e apri la ElastiCache console all'indirizzo <https://console.aws.amazon.com/elasticache/>.
2. Per visualizzare l'elenco dei cluster che eseguono Memcached, scegliere Memcached (Memcached) nel pannello di navigazione.

Per visualizzare un elenco dei cluster che eseguono Valkey o RedisOSS, nel pannello di navigazione scegli Valkey o Redis. OSS

3. Scegliere la casella a sinistra del cluster di cui modificare le notifiche.



4. Scegli Modifica.
5. In Modifica cluster in Argomento per la notifica, scegli Disattiva SNS notifiche.
6. Scegli Modifica.

#### Abilitazione e disabilitazione SNS delle notifiche Amazon ( )AWS CLI

Per disabilitare SNS le notifiche Amazon, usa 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
```

#### Abilitazione e disabilitazione SNS delle notifiche Amazon ( ) ElastiCache API

Per disabilitare SNS le notifiche Amazon, richiama l'`ModifyCacheCluster` azione 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 ElastiCache degli eventi

ElastiCache registra gli eventi relativi alle istanze del cluster, ai gruppi di sicurezza e ai gruppi di parametri. Queste informazioni includono la data, l'ora, il nome e tipo di fonte e una descrizione dell'evento. È possibile recuperare facilmente gli eventi dal registro utilizzando la ElastiCache console, il AWS CLI `describe-events` comando o l'azione. ElastiCache API `DescribeEvents`

Le procedure seguenti mostrano come visualizzare tutti ElastiCache gli eventi delle ultime 24 ore (1440 minuti).

### Visualizzazione ElastiCache degli eventi (Console)

La procedura seguente visualizza gli eventi utilizzando la ElastiCache console.

Per visualizzare gli eventi utilizzando la ElastiCache console

1. Accedi a AWS Management Console e apri la ElastiCache console 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 visualizza l'origine dell'evento, il tipo di evento (`cache-cluster`, `cache-parameter-group`, `cache-security-group`, or `cache-subnet-group`), l'GMTora dell'evento e una 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 ElastiCache eventi utilizzando il AWS CLI, utilizzare il comandod`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 ElastiCache degli eventi (ElastiCache API)

Per generare un elenco di ElastiCache eventi utilizzando il ElastiCache API, utilizzate l'DescribeEventsazione. 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 e Amazon SNS

ElastiCache può pubblicare messaggi utilizzando Amazon Simple Notification Service (SNS) quando si verificano eventi significativi su un cluster di cache. 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), incluse informazioni sui prezzi e link alla SNS documentazione di Amazon, consulta la [pagina SNS del prodotto Amazon](#).

Le notifiche vengono pubblicate su un SNS argomento Amazon specifico. Di seguito sono riportati i requisiti delle notifiche:

- È possibile configurare un solo argomento per ElastiCache le notifiche.
- L' AWS account proprietario dell'SNS argomento Amazon deve essere lo stesso account proprietario del cluster di cache su cui sono abilitate le notifiche.
- L'SNS argomento Amazon su cui stai pubblicando non può essere crittografato.

**Note**

È possibile allegare un SNS argomento Amazon crittografato (a riposo) al cluster. Tuttavia, lo stato dell'argomento dalla ElastiCache console verrà visualizzato come inattivo, il che di fatto dissocia l'argomento dal cluster quando ElastiCache invia messaggi all'argomento.

- L'SNS argomento Amazon deve trovarsi nella stessa regione del ElastiCache cluster.


## ElastiCache Eventi


I seguenti ElastiCache eventi attivano SNS le notifiche Amazon. Per ulteriori informazioni sui dettagli degli eventi, consulta [Visualizzazione ElastiCache degli eventi](#).

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 a causa di indirizzi IP gratuiti insufficienti	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 a causa di uno stato di rete incompatibile	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	È stato effettuato un tentativo di avviare un nuovo cluster di cache in un cloud privato virtuale inesistente (). VPC



Nome evento	Messaggio	Descrizione
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Dimensionamento di un cluster di cache completato.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</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"> <li>• L'elenco dei gruppi di sicurezza cache autorizzati per il cluster di cache è stato modificato.</li> <li>• Uno o più nuovi gruppi EC2 di sicurezza sono stati autorizzati su uno qualsiasi dei gruppi di sicurezza della cache associati al cluster di cache.</li> <li>• Uno o più gruppi EC2 di sicurezza sono stati revocati da uno qualsiasi dei gruppi di sicurezza della cache associati al cluster di cache.</li> </ul>

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 e ha iniziato a sostituire il nodo di cache.</p> <div data-bbox="1068 495 1507 810"><p> <b>Note</b></p><p>La DNS voce relativa al nodo di cache sostituito non viene modificata.</p></div> <p>Nella maggior parte dei casi, non è necessario aggiornar e l'elenco dei server per i client, quando si verifica questo evento. Tuttavia, alcune librerie client di cache potrebbero smettere di utilizzare il nodo di cache anche dopo ElastiCache aver sostituito il nodo di cache; in questo caso, l'applicazione dovrebbe aggiornare l'elenco dei server quando si verifica questo evento.</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 e ha completato la sostituzione del nodo di cache.</p> <div data-bbox="1068 541 1507 856"><p> <b>Note</b></p><p>La DNS voce relativa al nodo di cache sostituito non viene modificata.</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 di cache potrebbero smettere di utilizzare il nodo di cache anche dopo ElastiCache aver sostituito il nodo di cache; in questo caso, l'applicazione dovrebbe aggiornare l'elenco dei server quando si verifica questo evento.</p>

Nome evento	Messaggio	Descrizione
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Uno o più nodi di cache sono stati riavviati.  Messaggio (Memcached): "Cache node %s shutdown" Secondo messaggio: "Cache node %s restarted"
ElastiCache: CertificateRenewalComplete (solo Valkey o Redis) OSS	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 Valkey o Redis) OSS	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.

Nome evento	Messaggio	Descrizione
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	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.
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 <a href="#">Sostituzione dei nodi (Valkey e RedisOSS)</a> .
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 <a href="#">Sostituzione dei nodi (Valkey e RedisOSS)</a> .
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Un nodo di cache è stato rimosso dal cluster di cache.

Nome evento	Messaggio	Descrizione
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.
ElastiCache: SnapshotComplete (solo Valkey o Redis) OSS	ElastiCache:SnapshotComplete : <i>cluster-name</i>	È stata correttamente acquisita una snapshot di cache.
ElastiCache: SnapshotFailed (solo Valkey o Redis) OSS	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 <a href="#">DescribeSnapshots</a> , sarà failed.

## Argomenti correlati

- [Visualizzazione ElastiCache degli eventi](#)

# Consegna dei registri

## Note

Slow Log è supportato per Valkey 7.x e versioni successive e per i cluster di OSS cache e i gruppi di replica Redis che utilizzano il motore dalla versione 6.0 in poi.

Engine Log è supportato per Valkey 7.x e i cluster di OSS cache e i gruppi di replica Redis utilizzando la versione del motore 6.2 e successive.

La consegna dei log consente di eseguire lo streaming [SLOWLOG](#) Engine Log verso una delle due destinazioni seguenti:

- Amazon Data Firehose
- CloudWatch Registri Amazon

Puoi abilitare e configurare la consegna dei log quando crei o modifichi un cluster utilizzando ElastiCache APIs. Ogni voce di registro verrà recapitata alla destinazione specificata in uno dei due formati seguenti: JSON o TEXT.

Un numero fisso di voci Slow log viene recuperato periodicamente dal motore. 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 opzioni di modifica APIs, oppure [modify-cache-cluster](#). [modify-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

Lo 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 sostituisce il nome e il valore della chiave effettivi con `(2 more arguments)` per evitare l'esposizione di 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 ElastiCache del motore 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'UTCora 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 Primary, «S» per replica, «C» per Writer Child Process in lavorazione suRDB/AOFo «X» per Sentinel.
- Messaggio: messaggio di registro del motore.

## Autorizzazioni per configurare la registrazione

È necessario includere le seguenti IAM autorizzazioni nella politica IAM utente/ruolo:

- `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

Slow log supporta sia JSON TEXT

Di seguito viene illustrato un esempio di JSON formato:

```
{
 "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 viene illustrato un esempio di TEXT formato:

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

### Registro motore

Il registro del motore supporta JSON sia TEXT

Di seguito viene mostrato un esempio di JSON formato:

```
{
```

```
"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 viene illustrato un esempio di TEXT formato:

```
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 log di più OSS cluster Valkey o Redis e gruppi di replica possono essere consegnati allo stesso gruppo di log.
- 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 log su Logs CloudWatch

È necessario disporre delle seguenti impostazioni di autorizzazione per configurare l'invio di log ElastiCache 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

- È necessario specificare un flusso di distribuzione Firehose in cui verranno consegnati i log.

- I log provenienti da più OSS cluster Valkey o Redis e gruppi di replica possono essere inviati allo stesso flusso di distribuzione.
- 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 per configurare l'invio di log ElastiCache 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 il è AWS Management Console possibile creare un cluster Valkey o Redis OSS (modalità cluster disabilitata) seguendo i passaggi indicati [Creazione di un cluster Valkey \(modalità cluster disabilitata\) \(console\)](#) o creare un cluster Valkey o Redis OSS (modalità cluster abilitata) utilizzando i passaggi riportati in. [Creazione di un cluster Valkey o Redis OSS \(modalità cluster abilitata\) \(console\)](#) In entrambi i casi, è possibile configurare la consegna dei registri eseguendo le operazioni seguenti;

1. In Impostazioni avanzate, scegli Logs, quindi seleziona Slow logs o Engine logs.
2. In Formato registro, scegli Testo o. JSON
3. In Tipo di destinazione, scegli CloudWatch Logs o Kinesis Firehose.
4. In Destinazione del registro, scegli Crea nuovo e inserisci il nome del bucket Amazon S3, il nome del gruppo di CloudWatchLogs log o il nome dello stream Kinesis Data Firehose, oppure scegli Seleziona esistente e quindi scegli il nome del gruppo Logs o CloudWatch il nome dello stream 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. Accedi alla console e apri la console all'indirizzo. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Dal pannello di navigazione, scegli i cluster Valkey o i cluster Redis OSS.
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**, scegli **CloudWatch Logs** o **Kinesis Firehose**.
  - 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**.

## Specificare la consegna dei registri utilizzando il AWS CLI

### Registro lento

Crea 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 CloudWatch Metrics

ElastiCache fornisce metriche che consentono di monitorare i cluster. Puoi accedere a queste metriche tramite [CloudWatch](#). Per ulteriori informazioni su CloudWatch, consulta la [CloudWatch documentazione](#).

ElastiCache fornisce sia metriche a livello di host (ad esempio, CPU l'utilizzo) sia metriche specifiche del software del motore di cache (ad esempio, accessi e mancati accessi alla cache). Questi parametri vengono misurati e pubblicati per ogni nodo di cache in intervalli di 60 secondi.

**⚠ Important**

È consigliabile impostare CloudWatch allarmi su determinate metriche 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](#)
- [Metriche per Valkey e Redis OSS](#)
- [Parametri per Memcached](#)
- [Quali parametri è opportuno monitorare?](#)
- [Scelta delle statistiche e dei periodi di un parametro](#)
- [Monitoraggio delle metriche dei CloudWatch cluster e dei nodi](#)

**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**

- [Metriche per Valkey e Redis OSS](#)

Parametro	Descrizione	Unità
CPUUtilization	La percentuale di CPU utilizzo per l'intero host. Poiché Valkey e Redis OSS sono a thread singolo, consigliamo di monitorare la EngineCPUUtilization metrica per i nodi con 4 o più nodi. vCPUs	Percentuale
CPUCreditBalance	Il numero di CPU crediti guadagnati che un'istanza ha accumulato da quando è stata	Crediti (v - minutiCPU)

Parametro	Descrizione	Unità
	<p>lanciata o avviata. Per T2 Standard, include CPUCreditBalance anche il numero di crediti di lancio 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 inclusi in CPUCreditBalance possono essere spesi dall'istanza per superare il suo utilizzo di base. CPU</p> <p>CPUle metriche di credito sono disponibili solo con una frequenza di cinque minuti.</p> <p>Queste metriche non sono disponibili per le istanze di prestazioni Burstable T2.</p>	

Parametro	Descrizione	Unità
CPUCreditUsage	<p>Il numero di CPU crediti spesi dall'istanza per l'CPUUtilizzo. Un CPU credito equivale a una v che CPU funziona al 100% di utilizzo per un minuto o a una combinazione equivalente di vCPUs utilizzo e tempo (ad esempio, una v che CPU funziona al 50% di utilizzo per due minuti o due con un utilizzo del 25% per due vCPUs minuti).</p> <p>CPUle metriche di credito sono disponibili solo con una frequenza di cinque minuti. Se specifichi un periodo superiore a 5 minuti, usa la statistica Sum (somma) al posto di quella Average (media).</p> <p>Queste metriche non sono disponibili per le istanze di prestazioni Burstable T2.</p>	Crediti (v - minutiCPU)
FreeableMemory	La quantità di memoria libera disponibile sull'host. Questo è derivato dai RAM buffer e dalla cache che il sistema operativo riporta come liberabili.	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

Parametro	Descrizione	Unità
NetworkPacketsOut	Il numero di pacchetti inviati dall'istanza su tutte le interfacce di rete. Questo parametro identifica il volume del traffico in uscita in termini di numero di pacchetti su una singola istanza.	Conteggio
NetworkBandwidthInAllowanceExceeded	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 numero massimo di byte ricevuti al secondo in ogni minuto.	Byte
NetworkMaxBytesOut	Burst massimo al secondo di byte trasmessi in ogni minuto.	Byte
NetworkMaxPacketsIn	Il numero massimo di pacchetti ricevuti al secondo entro ogni minuto.	Conteggio

Parametro	Descrizione	Unità
NetworkMaxPacketsOut	Il numero massimo di pacchetti trasmessi al secondo in ogni minuto.	Conteggio
SwapUsage	La quantità di spazio di swapping utilizzato sull'host.	Byte

## Metriche per Valkey e Redis OSS

Il Amazon ElastiCache namespace include le seguenti metriche Valkey e Redis. OSS Queste metriche sono le stesse quando si utilizza il motore Valkey.

Ad eccezione di ReplicationLag e EngineCPUUtilization, queste metriche derivano dal comando info. Ogni parametro è calcolato a livello del nodo di cache.

Per la documentazione completa del info comando, vedere <http://valkey.io/commands/info>.

Vedi anche

- [Parametri a livello di host](#)

Parametro	Descrizione	Unità
ActiveDefragHits	Il numero di riallocazioni di valori al minuto effettuate dal processo di deframmentazione attivo. Questo è derivato dalla active_defrag_hits statistica di <a href="#">INFO</a> .	Numero
AuthenticationFailures	Il numero totale di tentativi falliti di autenticazione su Valkey o Redis OSS utilizzando il comando AUTH. È possibile trovare ulteriori informazioni sui singoli errori di autenticazione utilizzando il comando <a href="#">ACLLOG</a> . Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio



Parametro	Descrizione	Unità
	Il numero totale di byte allocati da Valkey o Redis OSS per tutti gli scopi, inclusi il set di dati, i buffer e così via.	Byte
BytesUsedForCache	Dimension: Tier=Memory per i OSS cluster Valkey o Redis utilizzando <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> : il numero totale di byte utilizzati per la cache in base alla memoria. Questo è il valore della statistica in. used_memory <a href="#">INFO</a>	Byte
	Dimension: Tier=SSD per i OSS cluster Valkey o Redis che utilizzano <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> : Il numero totale di byte utilizzati per la cache da. SSD	Byte
BytesReadFromDisk	Il numero totale di byte letti dal disco al minuto. Supportata solo per i cluster che utilizzano <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> .	Byte
BytesWrittenToDisk	Il numero totale di byte scritti su disco al minuto. Supportata solo per i cluster che utilizzano <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> .	Byte
CacheHits	Il numero di ricerche di chiavi di sola lettura nella directory principale. Questo è derivato dalla statistica di. keyspace_hits <a href="#">INFO</a>	Conteggio
CacheMisses	Il numero di ricerche di chiavi di sola lettura non riuscite nella directory principale. Questo è derivato dalla keyspace_misses statistica di. <a href="#">INFO</a>	Conteggio


Parametro	Descrizione	Unità
CommandAuthorizationFailures	Numero totale di tentativi non riusciti da parte degli utenti di eseguire comandi che non dispongono dell'autorizzazione per chiamare. È possibile trovare ulteriori informazioni sui singoli errori di autenticazione utilizzando il <a href="#">ACLLOG</a> comando. Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio
CacheHitRate	Indica l'efficienza di utilizzo dell'istanza Valkey o RedisOSS. 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. È possibile trovare ulteriori informazioni sui singoli errori di autenticazione utilizzando il comando. <a href="#">ACLLOG</a> Consigliamo di impostare un allarme su questa metrica per rilevare i tentativi di accesso non autorizzato.	Conteggio
CurrConnections	Il numero di connessioni client, escluse le connessioni dalle repliche di lettura. ElastiCache utilizza da due a quattro connessioni per monitorare il cluster in ogni caso. Questo è derivato dalla <code>connected_clients</code> statistica di <a href="#">INFO</a> .	Conteggio

Parametro	Descrizione	Unità
CurrItems	Il numero di elementi nella cache. Questo è derivato dalla keyspace statistica, sommando tutte le chiavi dell'intero spazio delle chiavi.	Conteggio
	Dimension: Tier=Memory per cluster che utilizzano <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> . Il numero di elementi in memoria.	Conteggio
	Dimension: Tier=SSD (Solid State Drive) per cluster che utilizzano <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> . Il numero di elementi in SSD	Conteggio
CurrVolatileItems	Numero totale di chiavi in tutti i database con un set ttl. Questo è derivato dalla expires statistica, sommando tutte le chiavi con un ttl impostato nell'intero spazio delle chiavi.	Conteggio
DatabaseCapacityUsagePercentage	<p>Percentuale della capacità di dati totale disponibile per il cluster in uso.</p> <p>Nelle istanze Data Tiered, la metrica viene calcolata come <math>(used\_memory - mem\_not\_counted\_for\_evict + SSD\ used) / (maxmemory + SSD\ total\ capacity)</math>, dove e da dove viene presa. <code>used_memory</code> <code>maxmemory</code> <a href="#">INFO</a></p> <p>In tutti gli altri casi, la metrica viene calcolata utilizzando <code>used_memory/maxmemory</code></p>	Percentuale

Parametro	Descrizione	Unità
DatabaseCapacityUsageCountedForEvictionPercentage	<p>Percentuale della capacità totale di dati per il cluster in uso, esclusa la memoria utilizzata per il sovraccarico e 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>da dove <code>maxmemory</code> vengono scattate <code>used_memory</code> e da cui sono state prese <a href="#">INFO</a></p>	Percentuale
DatabaseMemoryUsagePercentage	<p>Percentuale della memoria disponibile per il cluster in uso. Questo viene calcolato utilizzando <code>used_memory/maxmemory</code> da <a href="#">INFO</a>.</p>	Percentuale
DatabaseMemoryUsageCountedForEvictionPercentage	<p>Percentuale di memoria per il cluster in uso, esclusa la memoria utilizzata per il sovraccarico e COB. Viene calcolata utilizzando <code>used_memory - mem_not_counted_for_evict / maxmemory</code> from <a href="#">INFO</a>.</p>	Percentuale

Parametro	Descrizione	Unità
DB0AverageTTL	Esponde avg_ttl DBO dalla keyspace statistica del <a href="#">INFO</a> comando. 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 elimina a causa di essaLRU), sintetizza un DEL comando, che viene trasmesso a tutte le repliche. Pertanto, DB0Average TTL è 0 per i nodi di replica, poiché non fanno scadere le chiavi e quindi non ne tengono traccia. TTL	Millisecondi

Parametro	Descrizione	Unità
EngineCPUUtilization	Fornisce CPU l'utilizzo del thread del motore Valkey o Redis. OSS Poiché Valkey e Redis OSS sono a thread singolo, puoi utilizzare questa metrica per analizzare il carico del processo stesso. La EngineCPUUtilization metrica fornisce una visibilità più precisa del processo. Puoi usarla insieme alla metrica. CPUUtilization CPUUtilization espone CPU l'utilizzo dell'istanza del server nel suo complesso, inclusi altri sistemi operativi e processi di gestione. Per tipi di nodi più grandi con quattro vCPUs o più nodi, utilizza la EngineCPUUtilization metrica per monitorare e impostare soglie di scalabilità.	Percentuale


 Note

Su un ElastiCache host, i processi in background monitorano l'host per fornire un'esperienza di database gestita. Questi processi in background possono occupare una parte significativa del CPU carico di lavoro. Ciò non è significativo negli host più grandi con più di due vCPUs. Tuttavia, può influire sugli host più piccoli con 2 vCPUs o meno. Se monitorate solo la EngineCPUUtilization metrica, non sarete a conoscenza delle situazioni in cui l'host è sovraccarico sia a causa dell'elevato CPU utilizzo di Valkey o Redis OSS sia dell'elevato utilizzo dovuto ai processi di monitoraggio in CPU background. Pertanto, consigliamo di monitorare


Parametro	Descrizione	Unità
	la CPUUtilization metrica per gli host con due o meno. vCPUs	
Evictions	Il numero di chiavi che sono state rimosse a causa del limite maxmemory . Questo è derivato dalla evicted_keys statistica di. <a href="#">INFO</a>	Conteggio
GlobalDatastoreReplicationLag	Questo è il ritardo tra il nodo primario della regione secondaria e il nodo primario della regione primaria. Per Valkey o Redis abilitati in modalità clusterOSS, il ritardo indica il ritardo massimo tra gli shard.	Secondi
IamAuthenticationExpirations	Il numero totale di connessioni Valkey o Redis autenticate scaduteIAM. OSS Ulteriori informazioni su <a href="#">Autenticazione con IAM</a> sono disponibili nella guida per l'utente.	Conteggio
IamAuthenticationThrottling	Il numero totale di richieste o richieste Valkey o Redis con autenticazione limitataIAM. OSS AUTH HELLO Ulteriori informazioni su <a href="#">Autenticazione con IAM</a> sono disponibili nella guida per l'utente.	Conteggio
IsMaster	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. È possibile trovare ulteriori informazioni sui singoli errori di autenticazione utilizzando il comando. <a href="#">ACLLLOG</a> Consigliamo di impostare un allarme per rilevare i tentativi di accesso non autorizzato.	Conteggio
KeysTracked	Il numero di chiavi tracciate da Valkey o Redis OSS key tracking come percentuale di <code>tracking-table-max-keys</code> . Il tracciamento delle chiavi viene utilizzato per facilitare e la memorizzazione nella cache lato client e notifica ai client quando le chiavi vengono modificate.	Conteggio
MemoryFragmentationRatio	Indica l'efficienza nell'allocazione della memoria del motore Valkey o Redis. OSS Dalla soglia dipendono comportamenti diversi. Il valore consigliato è avere una frammentazione superiore a 1.0. Viene calcolato a partire da <code>mem_fragmentation_ratio</code> statistic <a href="#">INFO</a>	Numero



Parametro	Descrizione	Unità
NewConnections	<p>Il numero totale di connessioni accettate dal server durante questo periodo. Questo è derivato dalla <code>total_connections_received</code> statistica di <a href="#">INFO</a>.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Se utilizzi ElastiCache (RedisOSS) versione 5 o precedente, per monitorare e il cluster vengono utilizzate da due a quattro delle connessioni riportate ElastiCache da questa metrica. Tuttavia, quando si utilizza ElastiCache (RedisOSS) versione 6 o successiva, 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 <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> .	Conteggio
NumItemsWrittenToDisk	Il numero totale di elementi scritti su disco al minuto. Supportata solo per i cluster che utilizzano <a href="#">Suddivisione dei dati su più livelli in ElastiCache</a> .	Conteggio
MasterLinkHealthStatus	Questo stato ha due valori: 0 e 1. Il valore 0 indica che i dati nel nodo ElastiCache primario non sono sincronizzati con Valkey o Redis on. OSS EC2 Il valore 1 indica che i dati sono in fase di sincronizzazione. Per completare la migrazione, usa l' <a href="#">CompleteMigrationAPI</a> operazione.	Booleano

Parametro	Descrizione	Unità
Reclaimed	Il numero totale di eventi di scadenza di chiavi. Questo è derivato dalla <code>expired_keys</code> statistica di <a href="#">INFO</a> .	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. Questo è derivato dalla <code>master_repl_offset</code> statistica di <a href="#">INFO</a> .	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 Valkey 7.2 e versioni successive e il OSS motore Redis dalla versione 5.0.6 in poi, 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. <a href="#">INFO</a> Questo è derivato dalla statistica di <code>rdb_bgsave_in_progress</code> .	Booleano

Parametro	Descrizione	Unità
TrafficManagementActive	<p>Indica se ElastiCache (RedisOSS) gestisce attivamente il traffico regolando il traffico assegnato ai comandi in entrata, al monitoraggio o alla replica. Il traffico viene gestito quando al nodo vengono inviati più comandi di quanti ne possano elaborare Valkey o Redis OSS e viene utilizzato per mantenere la stabilità e il funzionamento ottimali 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> <b>Note</b></p> <p>Se questa metrica rimane attiva, valuta il cluster per decidere se è necessario aumentare o aumentare orizzontalmente. Le metriche correlate includono NetworkBandwidthOutAllowanceExceeded e EngineCPUUtilization .</p> </div>	Booleano

### Disponibilità di EngineCPUUtilization

AWS Le regioni elencate di seguito sono disponibili per tutti i tipi di nodi 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	US West (Oregon)

Regione	Nome Regione
ap-northeast-1	Asia Pacifico (Tokyo)
ap-northeast-2	Asia Pacifico (Seoul)
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 (San Paolo)

Regione	Nome Regione
us-gov-west-1	AWS GovCloud (Stati Uniti occidentali)
us-gov-east-1	AWS GovCloud (Stati Uniti orientali)

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 CPU tempo totale impiegato da questi comandi e la media CPU consumata per esecuzione del comando. Per ogni tipo di comando, viene aggiunta la seguente riga: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

Le metriche di latenza elencate di seguito sono calcolate utilizzando `commandstats statistic from`. **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 il CPU tempo impiegato per elaborare il comando. ElastiCache Tieni presente che per i cluster che utilizzano il tiering dei dati, il tempo impiegato per recuperare gli elementi da cui recuperare gli elementi non SSD è incluso in queste misurazioni.

Per un elenco completo dei comandi disponibili, consulta i [comandi](#) nella documentazione di Valkey.

Parametro	Descrizione	Unità
<code>ClusterBasedCmds</code>	Il numero totale di comandi basati su cluster. Questo è derivato dalla <code>commandstats</code> statistica sommando tutti i comandi che agiscono su un cluster ( <code>cluster slotcluster 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 evalsha</code>	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> . 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. Viene derivato dalla <code>commandstats</code> statistica sommando tutti i comandi di read-only tipo ( <code>get</code> , <code>hget</code> , <code>scardrange</code> , e così via).	Conteggio
GetTypeCmdsLatency	Latenza dei comandi di lettura.	Microsecondi
HashBasedCmds	Il numero totale di comandi basati su hash. Viene derivato dalla <code>commandstats</code> statistica sommando tutti i comandi che agiscono su uno o più hash ( <code>hget</code> , <code>hkeys</code> , <code>hvalshdel</code> , e così via).	Conteggio
HashBasedCmdsLatency	Latenza dei comandi basati su hash.	Microsecondi
HyperLogLogBasedCmds	Il numero totale di comandi basati su HyperLogLog. Viene ricavato dalla <code>commandstats</code> statistica sommando tutti i tipi di comandi ( <code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> , e così via).	Conteggio
HyperLogLogBasedCmdsLatency	Latenza dei comandi basati su HyperLogLog.	Microsecondi

Parametro	Descrizione	Unità
JsonBasedCmds	Il numero totale di JSON comandi, inclusi i comandi di lettura e scrittura. Viene ricavato dalla <code>commandstats</code> statistica sommando tutti i JSON comandi che agiscono sui JSON tasti.	Conteggio
JsonBasedCmdsLatency	Latenza di tutti i JSON comandi, inclusi i comandi di lettura e scrittura.	Microsecondi
JsonBasedGetCmds	Il numero totale di comandi di JSON sola lettura. Viene ricavato dalla <code>commandstats</code> statistica sommando tutti i comandi di JSON lettura che agiscono sui tasti. JSON	Conteggio
JsonBasedGetCmdsLatency	Latenza dei comandi di sola letturaJSON.	Microsecondi
JsonBasedSetCmds	Il numero totale di comandi di scrittura. JSON Questo è derivato dalla <code>commandstats</code> statistica sommando tutti i comandi di JSON scrittura che agiscono sui JSON tasti.	Conteggio
JsonBasedSetCmdsLatency	Latenza dei comandi di scrittura. JSON	Microsecondi
KeyBasedCmds	Il numero totale di comandi basati su chiavi. Questa è derivata dalla <code>commandstats</code> statistica sommando tutti i comandi che agiscono su una o più chiavi su più strutture di dati ( <code>del</code> , <code>expire</code> , <code>rename</code> , e così via).	Conteggio
KeyBasedCmdsLatency	Latenza dei comandi basati su chiave.	Microsecondi

Parametro	Descrizione	Unità
ListBasedCmds	Il numero totale di comandi basati su elenchi. Questo valore è derivato dalla <code>commandstats</code> statistica sommando tutti i comandi che agiscono su uno o più elenchi ( <code>lindex</code> , <code>lrange</code> <code>lpushltrim</code> , e così via).	Conteggio
ListBasedCmdsLatency	Latenza dei comandi basati su elenchi.	Microsecondi
NonKeyTypeCmds	Il numero totale di comandi non basati su chiavi. Questo valore è derivato dalla <code>commandstats</code> statistica sommando tutti i comandi che non agiscono su una chiave, ad esempio, <code>o. acl dbsize info</code>	Conteggio
NonKeyTypeCmdsLatency	Latenza dei comandi. non-key-based	Microsecondi
PubSubBasedCmds	Numero totale di comandi per la funzionalità pub/sub. Questa è derivata dalle <code>commandstats</code> statistiche sommando tutti i comandi utilizzati per la funzionalità pub/sub: <code>punsubscribe</code> , <code>publish</code> , <code>pubsubpunsubscribe</code> , <code>unsubscribe</code> , <code>sunsubscribe</code> e <code>spublish subscribe unsubscribe</code>	Conteggio
PubSubBasedCmdsLatency	Latenza dei comandi pub/sub-based.	Microsecondi
SetBasedCmds	Il numero totale di comandi basati su set. Questo è derivato dalla <code>commandstats</code> statistica sommando tutti i comandi che agiscono su uno o più set ( <code>scard</code> , <code>sdiffsadd</code> , <code>union</code> e così via).	Conteggio
SetBasedCmdsLatency	Latenza dei comandi basati su set.	Microsecondi



Parametro	Descrizione	Unità
SetTypeCmds	Il numero totale di comandi di tipo write. Questo valore è derivato dalla <code>commandstats</code> statistica sommando tutti i mutative tipi di comandi che operano sui dati ( <code>set</code> , <code>hset</code> , <code>sadd</code> , <code>pop</code> , e così via).	Conteggio
SetTypeCmdsLatency	Latenza dei comandi di scrittura.	Microsecondi
SortedSetBasedCmds	Il numero totale di comandi basati su set ordinati. Viene derivato dalla <code>commandstats</code> statistica 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
SortedSetBasedCmdsLatency	Latenza dei comandi basati su ordinazione.	Microsecondi
StringBasedCmds	Il numero totale di comandi basati su stringhe. Viene derivato dalla <code>commandstats</code> statistica sommando tutti i comandi che agiscono su una o più stringhe ( <code>strlen</code> , <code>setex</code> , <code>setrange</code> e così via).	Conteggio
StringBasedCmdsLatency	Latenza dei comandi basati su stringhe.	Microsecondi
StreamBasedCmds	Il numero totale di comandi basati sul flusso. Viene derivato dalla <code>commandstats</code> statistica sommando tutti i comandi che agiscono su uno o più tipi di dati di stream ( <code>xrange</code> , <code>xlen</code> , <code>xadd</code> , <code>xdel</code> e così via).	Conteggio
StreamBasedCmdsLatency	Latenza dei comandi basati sul flusso.	Microsecondi

## Parametri per Memcached

Il namespace `AWS/ElastiCache` include i parametri Memcached descritti di seguito.

Lo spazio dei nomi `ElastiCache` include le seguenti metriche derivate dal comando `Memcached stats`. Ogni parametro è calcolato a livello del nodo di cache.

Consulta anche

- [Parametri a livello di host](#)

Parametro	Descrizione	Unità
<code>BytesReadIntoMemcached</code>	Il numero di byte che il nodo di cache ha letto dalla rete.	Byte
<code>BytesUsedForCacheItems</code>	Il numero di byte utilizzati per archiviare le voci di cache.	Byte
<code>BytesWrittenOutFromMemcached</code>	Il numero di byte che il nodo di cache ha scritto sulla rete.	Byte
<code>CasBadval</code>	Il numero di richieste CAS (check and set) ricevute dalla cache in cui il valore Cas non corrisponde al valore Cas memorizzato.	Conteggio
<code>CasHits</code>	Il numero di richieste Cas che la cache ha ricevuto dove la chiave richiesta è stata trovata e il valore Cas corrispondeva.	Conteggio
<code>CasMisses</code>	Il numero di richieste Cas che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
<code>CmdFlush</code>	Il numero di comandi flush che la cache ha ricevuto.	Conteggio
<code>CmdGet</code>	Il numero di comandi get che la cache ha ricevuto.	Conteggio

Parametro	Descrizione	Unità
<code>CmdSet</code>	Il numero di comandi stabiliti che la cache ha ricevuto.	Conteggio
<code>CurrConnections</code>	<p>Il conteggio del numero di connessioni connesse alla cache in un dato momento. ElastiCache utilizza da due a tre connessioni per monitorare il cluster.</p> <p>Memcached crea inoltre un numero di connessioni interne pari al doppio dei thread utilizzati per il tipo di nodo. Il conteggio dei thread per i vari tipi di nodo può essere osservato nel gruppo di parametri applicabile <code>Nodetype Specific Parameters</code> .</p> <p>Le connessioni totali sono la somma delle connessioni client, delle connessioni per il monitoraggio e delle connessioni interne citate sopra.</p>	Conteggio
<code>CurrItems</code>	Il conteggio del numero di elementi attualmente memorizzati nella cache.	Conteggio
<code>DecrHits</code>	Il numero di richieste di decremento che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio
<code>DecrMisses</code>	Il numero di richieste di decremento che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
<code>DeleteHits</code>	Il numero di richieste di eliminazione che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio

Parametro	Descrizione	Unità
DeleteMisses	Il numero di richieste di eliminazione che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
Evictions	Il numero di elementi non scaduti che la cache ha rimosso per creare spazio per nuove scritture.	Conteggio
GetHits	Il numero di richieste di recupero che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio
GetMisses	Il numero di richieste di recupero che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
IncrHits	Il numero di richieste di incremento che la cache ha ricevuto dove la chiave richiesta è stata trovata.	Conteggio
IncrMisses	Il numero di richieste di incremento che la cache ha ricevuto dove la chiave richiesta non è stata trovata.	Conteggio
Reclaimed	Il numero di elementi scaduti che la cache ha rimosso per creare spazio per nuove scritture.	Conteggio

Per Memcached 1.4.14 vengono forniti i seguenti parametri aggiuntivi.

Parametro	Descrizione	Unità
BytesUsedForHash	Il numero di byte attualmente utilizzati dalle tabelle hash.	Byte
CmdConfigGet	Il numero cumulativo di richieste config get.	Conteggio

Parametro	Descrizione	Unità
CmdConfigSet	Il numero cumulativo di richieste config set.	Conteggio
CmdTouch	Il numero cumulativo di richieste touch.	Conteggio
CurrConfig	Il numero corrente di configurazioni memorizzate.	Conteggio
EvictedUnfetched	Il numero di elementi validi rimossi dalla cache (LRU) usata meno di recente che non sono mai stati toccati dopo essere stati impostati.	Conteggio
ExpiredUnfetched	Il numero di elementi scaduti recuperati e LRU che non sono mai stati toccati dopo essere stati impostati.	Conteggio
SlabsMoved	Il numero totale di pagine di allocazione memoria che sono state spostate.	Conteggio
TouchHits	Il numero di chiavi che sono state toccate e a cui è stata assegnata una nuova ora di scadenza.	Conteggio
TouchMisses	Il numero di elementi che sono stati toccati, ma non trovati.	Conteggio

Lo ElastiCache spazio dei nomi AWS/include le seguenti metriche calcolate a livello di cache.

Parametro	Descrizione	Unità
NewConnections	Il numero di nuove connessioni che la cache ha ricevuto. Tale numero viene ricavato dalla statistica Memcached total_connections mediante la registrazione della modifica in total_connections per un periodo di tempo. Questo sarà sempre almeno 1, a causa di una connessione riservata a. ElastiCache	Conteggio

Parametro	Descrizione	Unità
NewItems	Il numero di nuovi elementi che la cache ha memorizzato. Tale numero viene ricavato dalla statistica Memcached total_items mediante la registrazione della modifica in total_items per un periodo di tempo.	Conteggio
UnusedMemory	<p>La quantità di memoria non utilizzata dai dati. Tale numero viene ricavato dalla statistica Memcached limit_maxbytes e bytes sottraendo bytes da limit_maxbytes.</p> <p>Poiché l'overhead di Memcached utilizza memoria in aggiunta a quella utilizzata dai dati, non UnusedMemory deve essere considerata la quantità di memoria disponibile per dati aggiuntivi. Potrebbero verificarsi delle rimozioni anche se hai ancora della memoria inutilizzata.</p> <p>Per ulteriori informazioni, consulta <a href="#">Utilizzo della memoria degli elementi Memcached</a>.</p>	Byte

## Quali parametri è opportuno monitorare?

Le seguenti CloudWatch metriche offrono una buona visione ElastiCache delle prestazioni. Nella maggior parte dei casi, ti consigliamo di impostare CloudWatch allarmi per queste metriche in modo da poter intraprendere azioni correttive prima che si verifichino problemi di prestazioni.

### Parametri da monitorare

- [CPUUtilization](#)
- [EngineCPUUtilization](#)
- [SwapUsage \(Valkey e Redis\) OSS](#)
- [Espulsioni](#)
- [CurrConnections](#)
- [Memoria \(Valkey e Redis\) OSS](#)
- [Rete](#)
- [Latenza](#)
- [Replica](#)
- [Gestione del traffico \(Valkey e Redis\) OSS](#)

### CPUUtilization

Si tratta di un parametro a livello di host restituito sotto forma di percentuale. Per ulteriori informazioni, consulta [Parametri a livello di host](#).

### Valkey e Redis OSS

Per tipi di nodi più piccoli con 2 vCPUs o meno, utilizza la `CPUUtilization` metrica per monitorare il carico di lavoro.

In generale, ti suggeriamo di impostare la soglia al 90% del valore disponibile. CPU Poiché Valkey e Redis OSS sono entrambi 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 Valkey o Redis OSS (modalità cluster disabilitata): scalabilità verticale utilizzando un tipo di istanza di cache più grande.
- Cluster Valkey o Redis OSS (modalità cluster abilitata): aggiungi altri shard per distribuire il carico di lavoro di scrittura su più nodi primari.

#### Tip

Invece di utilizzare la metrica a livello di `hostCPUUtilization`, OSS gli utenti di Valkey e Redis potrebbero utilizzare la metrica, che riporta la percentuale di utilizzo sul core del motore `EngineCPUUtilization` Valkey o Redis. OSS [Per vedere se questa metrica è disponibile sui tuoi nodi e per ulteriori informazioni, consulta Metrics for Valkey e Redis. OSS](#)

Per tipi di nodi più grandi con 4 vCPUs o più, potresti voler utilizzare la `EngineCPUUtilization` metrica, che riporta la percentuale di utilizzo sul core del motore Valkey o Redis. OSS [Per vedere se questa metrica è disponibile sui tuoi nodi e per ulteriori informazioni, consulta Metrics for Redis. OSS](#)

## Memcached

Poiché Memcached è multi-thread, questo parametro può arrivare a 90%. Se superi questa soglia, espandi il cluster di cache utilizzando un tipo di nodo di cache più grande o ridimensionalo aggiungendo altri nodi di cache.

## EngineCPUUtilization

Per tipi di nodi più grandi con 4 vCPUs o più, potresti voler utilizzare la `EngineCPUUtilization` metrica, che riporta la percentuale di utilizzo sul core del OSS motore Redis. Per vedere se questa metrica è disponibile sui tuoi nodi e per ulteriori informazioni, consulta [Metriche per](#) Valkey e Redis. OSS

Per ulteriori informazioni, consulta la CPU sezione [Monitoraggio delle best practice con Amazon ElastiCache \(RedisOSS\) con Amazon CloudWatch](#).



## SwapUsage (Valkey e Redis) OSS

Si tratta di un parametro a livello di host restituito in byte. Per ulteriori informazioni, consulta [Parametri a livello di host](#).

Se la `FreeableMemory` CloudWatch metrica è vicina a 0 (ovvero inferiore a 100 MB) o una `SwapUsage` metrica maggiore della `FreeableMemory` metrica indica che un nodo è sotto pressione in termini di memoria. In tal caso, consulta i seguenti argomenti:

- [Assicurarsi di disporre di memoria sufficiente per creare un'istanza Valkey o Redis OSS](#)
- [Gestione della memoria riservata per Valkey e Redis OSS](#)

## 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.

Se utilizzi Memcached e superi la soglia prescelta, amplia il cluster utilizzando un tipo di nodo più grande o scalalo orizzontalmente aggiungendo altri nodi.

## 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 dati `CurrConnections` potrebbe indicare un problema con l'applicazione; per risolvere il problema, sarà necessario esaminare il comportamento dell'applicazione.

Per ulteriori informazioni, consulta la sezione Connessioni in [Monitoring best practice with Amazon ElastiCache \(RedisOSS\) using Amazon CloudWatch](#).

## Memoria (Valkey e Redis) OSS

La memoria è un aspetto fondamentale di Valkey e Redis. OSS 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 memoria del comando.

### [INFO](#)

Per ulteriori informazioni, consulta la sezione Memoria in [Monitoring best practice with Amazon ElastiCache \(RedisOSS\) using 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 tuo nodo, consulta [ElastiCache i prezzi di Amazon](#).

Per ulteriori informazioni, consulta la sezione Rete in [Monitoring best practice with Amazon ElastiCache \(RedisOSS\) using Amazon CloudWatch](#).

## Latenza

Puoi misurare la latenza di un comando con un set di CloudWatch metriche che forniscono latenze aggregate per struttura di dati. Queste metriche di latenza vengono calcolate utilizzando la statistica del `commandstats` comando Valkey. [INFO](#)

Per ulteriori informazioni, consulta la sezione Latenza in [Monitoring best practice with Amazon ElastiCache using 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 in [Monitoring best practice with Amazon ElastiCache \(RedisOSS\) using Amazon CloudWatch](#).

## Gestione del traffico (Valkey e Redis) OSS

ElastiCache (RedisOSS) gestisce automaticamente il traffico verso un nodo quando al nodo vengono inviati più comandi in entrata di quanti ne possano essere elaborati da Valkey o Redis. OSS 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 ti consenta di scegliere qualsiasi statistica e periodo per ogni metrica, non tutte le combinazioni saranno utili. Ad esempio, le statistiche Media, Minima e Massima per CPUUtilization sono utili, ma la statistica Sum no.

Tutti gli ElastiCache esempi 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 delle metriche dei CloudWatch cluster e dei nodi](#).

## Monitoraggio delle metriche dei CloudWatch cluster e dei nodi

ElastiCache e CloudWatch sono integrati in modo da poter raccogliere una varietà di metriche. È possibile monitorare queste metriche utilizzando CloudWatch

### Note

Gli esempi seguenti richiedono gli strumenti della CloudWatch riga di comando. Per ulteriori informazioni CloudWatch e per scaricare gli strumenti per sviluppatori, consulta la [pagina CloudWatch del prodotto](#).

Le seguenti procedure mostrano come CloudWatch raccogliere le statistiche sullo spazio di archiviazione per un cluster 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 ElastiCache limiti, vedere [AWS Service Limits](#) for ElastiCache.

## Monitoraggio delle metriche dei CloudWatch cluster e dei nodi (console)

Per raccogliere statistiche CPU di utilizzo per un cluster di cache

1. Accedi a AWS Management Console e apri la ElastiCache console 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 Cluster di cache della Console di AWS gestione, fai 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 un elenco di CloudWatch metriche disponibili.

- d. Fai clic sulla metrica CPU di utilizzo.

La CloudWatch console si aprirà e mostrerà le metriche selezionate. È possibile modificare i parametri visualizzati, mediante gli elenchi a discesa di Statistic (Statistica) e Period (Periodo) e la scheda Time Range (Intervallo di tempo).

## Monitoraggio delle metriche del CloudWatch cluster e dei nodi utilizzando il CloudWatch CLI

Per raccogliere statistiche CPU di utilizzo per 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 CloudWatch cluster e dei nodi utilizzando il CloudWatch API

Per raccogliere statistiche CPU di utilizzo per un cluster di cache

- Chiamali CloudWatch API GetMetricStatistics con i seguenti parametri (nota che gli orari di inizio e fine sono mostrati solo a titolo di esempio; dovrai sostituirli con gli orari di inizio e di fine 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>
```

## Quote per ElastiCache

Il tuo AWS account ha delle quote predefinite, precedentemente denominate limiti, per ogni servizio. AWS Salvo diversa indicazione, ogni quota si applica a una regione specifica. Se per alcune quote è possibile richiedere aumenti, altre quote non possono essere modificate.

Per visualizzare le quote per ElastiCache, apri la console [Service Quotas](#). Nel riquadro di navigazione, scegli AWS servizi e seleziona. 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 AWS account ha le seguenti quote relative a ElastiCache.

Risorsa	Di default
Cache serverless per regione	40
Istantanee serverless al giorno per cache, Redis	24
Nodi per regione	300
Nodi per cluster, Memcached	60
Nodi per cluster per tipo di istanza, Valkey o Redis OSS (modalità cluster abilitata)	90
Nodi per shard, Valkey o Redis OSS (modalità cluster 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

Risorsa	Di default
Utenti per gruppo di utenti, Redis	100
Numero massimo di utenti, Redis	1000
Numero massimo di gruppi di utenti, Redis	100



# Riferimento

Gli argomenti di questa sezione riguardano l'utilizzo di Amazon ElastiCache API e la ElastiCache sezione di AWS CLI. Sono inclusi anche i messaggi di errore e le notifiche di servizio più frequenti.

- [Usando il ElastiCache API](#)
- [ElastiCache API Riferimento](#)
- [ElastiCache sezione del AWS CLI Reference](#)
- [Messaggi ElastiCache di errore Amazon](#)
- [Notifiche](#)

## Usando il ElastiCache API

Questa sezione fornisce descrizioni orientate alle attività su come utilizzare e implementare le operazioni. ElastiCache Per una descrizione completa di queste operazioni, consulta [Amazon ElastiCache API Reference](#).

### Argomenti

- [Utilizzo dell'interrogazione API](#)
- [Librerie disponibili](#)
- [Risoluzione dei problemi delle applicazioni](#)

## Utilizzo dell'interrogazione API

### Parametri di query

HTTPLe richieste basate su query sono HTTP richieste che utilizzano il HTTP verbo GET or POST e un parametro 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`. Valori di `n` sono numeri interi a partire da 1.

## Autenticazione delle richieste di query

È possibile inviare solo richieste di Query 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. Ciò presuppone che tu sia registrato 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 a AWS
2. Il mittente calcola la firma della richiesta, un codice di autenticazione dei messaggi basato su Keyed-Hashing for Hash (HMAC) con una funzione hash SHA -1, come definito nella sezione successiva di questo argomento.
3. Il mittente della richiesta invia i dati della richiesta, la firma e l'ID della chiave di accesso (l'identificatore della chiave di accesso segreta utilizzata) a AWS
4. AWS utilizza l'ID della chiave di accesso per cercare la chiave di accesso segreta.
5. AWS genera una firma dai dati della richiesta e dalla chiave di accesso segreta utilizzando lo stesso algoritmo utilizzato per calcolare 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. Ordina i componenti della stringa di query UTF -8 in base al nome del parametro con un ordinamento naturale dei byte. I parametri possono provenire dal GET URI o dal POST corpo (quando Content-Type è application/). x-www-form-urlencoded
  - b. URLcodifica il nome e i valori dei parametri secondo le seguenti regole:
    - i. Non URL codificate nessuno dei caratteri non riservati definiti da 3986. RFC 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. Codifica percentuale estesa a UTF -8 caratteri nel formato %XY%ZA...
    - iv. Codificare con codifica percentuale il carattere di spazio come %20 (non utilizzare +, come negli schemi di codifica comuni).
  - c. Separa i nomi dei parametri codificati dai relativi valori codificati con il segno di uguale (=) (ASCIIcarattere 61), anche se il valore del parametro è vuoto.
  - d. Separate le coppie nome-valore con una e commerciale (&) (codice 38). ASCII
2. Create la stringa da firmare secondo la seguente pseudo-grammatica ("`\n`" rappresenta una nuova riga). ASCII

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

Il HTTPRequestURI componente è il componente del percorso HTTP assoluto della stringa di query URI fino a, ma non include. Se HTTPRequestURI è vuoto, usa una barra (/).

3. Calcola un valore RFC HMAC conforme a 2104 utilizzando la stringa appena creata, la chiave di accesso segreta come chiave SHA256 e/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, dovresti calcolare la firma sulla HMAC stringa seguente.

```
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 fornisce kit di sviluppo software (SDKs) per gli sviluppatori di software che preferiscono creare applicazioni utilizzando specifiche lingue APIs anziché Query. API Questi SDKs forniscono funzioni di base (non incluse in APIs), come l'autenticazione delle richieste, i nuovi tentativi di richiesta e la gestione degli errori, in modo che sia più facile iniziare. SDKse sono disponibili 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 facilitare la risoluzione dei problemi durante l'interazione con. ElastiCache API

### 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 scoprire se si è verificato un errore consiste nel cercare un `ERROR` nodo nella risposta di. ElastiCache API

XPathla sintassi fornisce un modo semplice per cercare la presenza di un `ERROR` nodo, nonché un modo semplice per recuperare il codice e il messaggio di errore. Il seguente frammento di codice utilizza Perl e il XPath moduloXML:: 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

Consigliamo i seguenti processi per diagnosticare e risolvere i problemi con. ElastiCache API

- Verifica che ElastiCache funzioni correttamente.

Per fare ciò, è sufficiente aprire una finestra del browser e inviare una richiesta di query al ElastiCache servizio (ad esempio <https://elasticache.amazonaws.com>). Un `MissingAuthenticationTokenException` errore interno del server pari a 500 conferma che il servizio è disponibile e risponde alle richieste.

- Verificare la struttura della richiesta.

Ogni ElastiCache operazione ha una pagina di riferimento nel ElastiCache APIReference. 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.

ElastiCache dispone di un forum di discussione in cui è possibile cercare soluzioni ai problemi che altri hanno riscontrato lungo il percorso. Per visualizzare il forum, accedi all'indirizzo

<https://forums.aws.amazon.com/> .

## Configurazione dell'interfaccia ElastiCache a riga di comando

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.

Segui le istruzioni riportate in questo argomento solo se intendi utilizzare AWS CLI la ElastiCache forchetta.

### Important

Amazon ElastiCache Command Line Interface (CLI) non supporta alcun ElastiCache miglioramento dopo la API versione 2014-09-30. [Per utilizzare nuove ElastiCache funzionalità dalla riga di comando, usa l'interfaccia a riga di comando.AWS](#)

## Argomenti

- [Prerequisiti](#)
- [Ottenere gli strumenti a riga di comando](#)
- [Configurazione degli strumenti](#)
- [Fornire le credenziali per gli strumenti](#)
- [Variabili di ambiente](#)

## Prerequisiti

Questo documento presuppone che si possa lavorare in un ambiente UNIX Linux/ o Windows. Gli strumenti a riga di ElastiCache comando di Amazon funzionano anche su Mac OS X, che è un ambiente UNIX basato su Mac OS X; tuttavia, in questa guida non sono incluse istruzioni specifiche per Mac OS X.

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 per Linux/ e **C:\>** per un comando UNIX specifico per 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. Un'JDKinstallazione JRE o è accettabile. Per visualizzare e scaricare JREs per un'ampia gamma di piattaforme, tra cui Linux/ UNIX e Windows, consulta [Java SE Downloads](#).

### 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. Questa variabile di ambiente deve essere impostata sul percorso

completo della directory che contiene una sottodirectory denominata `bin` che a sua volta contiene l'eseguibile `java` (su Linux eUNIX) o `java.exe` (su Windows) eseguibile.

Per impostare la variabile Java Home

#### 1. Impostare la variabile Java Home.

- Su Linux eUNIX, inserisci 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, vedrete un output 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 da riga di comando sono disponibili come ZIP file sul sito web [ElastiCache Developer Tools](#). Questi strumenti sono scritti in Java e includono script di shell per Windows 2000/XP/Vista/



Windows 7, Linux/ e Mac. UNIX OSX Il ZIP file è autonomo e non è richiesta alcuna installazione; è sufficiente scaricare il file zip e decomprimerlo in una directory sul computer locale.

## Configurazione degli strumenti

Gli strumenti da riga di comando dipendono da una variabile di ambiente (`AWS_ELASTICACHE_HOME`) per individuare le librerie di supporto. 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. Questa directory è denominata `ElastiCacheCli -A.B.nnnn` (A, B e n sono numeri di versione/release) e contiene le sottodirectory denominate `bin` e `lib`.

Per `ELASTICACHE` impostare la `HOME` variabile di ambiente `__AWS`

- Aprire una finestra della riga di comando e immettere uno dei seguenti comandi per impostare la variabile di HOME ambiente `AWS ELASTICACHE __`.
- Su Linux eUNIX, immettete 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 rendere gli strumenti più facili da usare, ti consigliamo di aggiungere la `BIN` cartella degli strumenti al tuo sistema `PATH`. Il resto di questa guida presuppone che la `BIN` directory si trovi nel percorso di sistema.

Per aggiungere la `BIN` cartella degli strumenti al percorso di sistema

- Immettete i seguenti comandi per aggiungere la `BIN` cartella degli strumenti al sistema. `PATH`
- Su Linux eUNIX, inserisci 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

Gli strumenti da riga di comando richiedono la chiave di AWS accesso e la chiave di accesso segreta fornite con l' AWS account. Puoi ottenerle tramite la riga di comando o da un file di credenziali presente nel sistema locale.

La distribuzione include un file modello `{AWS_ ELASTICACHE _HOME}/credential-file-path.template` che devi modificare con le tue informazioni. Di seguito i contenuti del file di modello:

```
AWS AccessKeyId=<Write your AWS access ID>
AWS SecretKey=<Write your AWS secret key>
```

**Important**

SìUNIX, limita le autorizzazioni al proprietario del file di credenziali:

```
$ chmod 600 <the file created above>
```

Con la configurazione del file delle credenziali, dovrai impostare la variabile di FILE ambiente `AWS_CREDENTIAL_` in modo che ElastiCache gli strumenti possano trovare le tue informazioni.

Per impostare la variabile di FILE ambiente `AWS_CREDENTIAL_`

## 1. Imposta la variabile di ambiente :

- Su Linux eUNIX, aggiorna la variabile usando il seguente comando:

```
$ 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
```

Dovresti vedere la pagina di utilizzo per tutti i ElastiCache comandi.

## 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 FILE ambiente `AWS_CREDENTIAL_`, la maggior parte API degli strumenti inclusi nell'interfaccia a riga di ElastiCache comando supporta le seguenti variabili:

- `EC2_REGION` — La AWS regione da utilizzare.
- `AWS_ELASTICACHE_URL` — URL Da utilizzare per la chiamata di servizio. Non è necessario specificare un endpoint regionale diverso se `REGION` viene specificato `EC2_` o viene passato il parametro `--region`.

Gli esempi seguenti mostrano come impostare la variabile di ambiente `EC2_REGION` per configurare la regione utilizzata dagli API strumenti:

Linux, OS X o Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

## Messaggi ElastiCache di errore Amazon

I seguenti messaggi di errore vengono restituiti da Amazon ElastiCache. Potresti ricevere altri messaggi di errore restituiti da ElastiCache altri AWS servizi o da Valkey, Redis o OSS Memcached. Per le descrizioni dei messaggi di errore provenienti da fonti diverse ElastiCache, consulta la documentazione della fonte che genera il messaggio di errore.

- [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. Oppure, se hai bisogno di più di %n nodi, effettua la richiesta utilizzando il [modulo di richiesta Amazon ElastiCache Node](#).

Per ulteriori informazioni, consulta [Amazon ElastiCache Limits](#) in 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. Oppure, se hai bisogno di più di %n nodi, effettua la richiesta utilizzando il [modulo di richiesta Amazon ElastiCache Node](#).

Per ulteriori informazioni, consulta [Amazon ElastiCache Limits](#) in 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.

Messaggi 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 [Ridimensionamento ElastiCache](#).

## Notifiche

Questo argomento riguarda ElastiCache le notifiche che potrebbero interessarti. 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 ElastiCache caratteristiche o funzionalità nuove o migliorate.

## ElastiCache Notifiche generali

Al momento non ci sono ElastiCache notifiche in sospeso che non siano specifiche del motore.


## ElastiCache Notifiche (Memcached)

Le seguenti ElastiCache notifiche sono specifiche del motore Memcached.

ElastiCache Notifiche specifiche (Memcached)

- [Avviso: il crawler Memcached LRU causa errori di segmentazione](#)

### Avviso: il crawler Memcached LRU causa errori di segmentazione

 Data di avviso: 28 febbraio 2017

In alcune circostanze, il cluster potrebbe mostrare instabilità con un errore di segmentazione nel Crawler Memcached. LRU Si tratta di un problema nel motore Memcached presente per un periodo di tempo. Il problema è diventato evidente in Memcached 1.4.33 quando il Crawler era abilitato per impostazione predefinita. LRU

Se riscontri questo problema, ti consigliamo di disabilitare il LRU Crawler fino a quando non viene trovata una soluzione. A tale scopo, utilizza `lru_crawler disable` nella riga di comando o modifica il valore del parametro `lru_crawler` (preferito).

Data di risoluzione:

Risoluzione:

## ElastiCache (RedisOSS) notifiche specifiche

Al momento non ci sono notifiche ElastiCache (RedisOSS) in sospeso.

# ElastiCache Cronologia della documentazione

- APIversione: 2015-02-02
- Ultimo aggiornamento della documentazione: 27 novembre 2023

La tabella seguente descrive le modifiche importanti apportate in ogni versione della Guida per l'ElastiCache utente dopo marzo 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi iscriverti al RSS feed.

## ElastiCache Aggiornamenti recenti

Modifica	Descrizione	Data
<a href="#">Support per ElastiCache con Valkey</a>	ElastiCache ora supporta Valkey. <a href="#">Valkey 7.2.6 è compatibile con Redis OSS 7.2</a> Per ulteriori informazioni, <a href="#">consulta Valkey</a> .	8 ottobre 2024
<a href="#">Dimensioni dei nodi riservati flessibili</a>	ElastiCache ora supporta i <a href="#">nodi riservati Size flexible</a> . Per ulteriori informazioni, consulta la pagina <a href="#">ElastiCache dei prezzi di Amazon</a> .	1 ottobre 2024
<a href="#">ElastiCache (Redis) ha aggiunto il supporto per nodi C7gn di OSS dimensioni aggiuntive</a>	ElastiCache (RedisOSS) ha aggiunto il supporto per nodi C7gn di dimensioni aggiuntive.	10 gennaio 2024
<a href="#">ElastiCache (RedisOSS) ora supporta la creazione di cache serverless</a>	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,	27 novembre 2023

consulta [Scelta tra le opzioni di implementazione](#). Come parte di questa funzionalità, sono state aggiunte [nuove autorizzazioni](#) `ElastiCacheServiceRolePolicy` e per consentire l'associazione di `AmazonElastiCacheFullAccess` `cache` serverless con endpoint gestiti. VPC Inoltre, sono state aggiunte le autorizzazioni per supportare un'esperienza di console modificata utilizzando la policy `AmazonElastiCacheFullAccess`.



[ElastiCache \(Memcached\) 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](#) `ElastiCacheServiceRolePolicy` e `AmazonElastiCacheFullAccess` per consentire l'associazione di cache serverless con endpoint gestiti. VPC Inoltre, sono state aggiunte le autorizzazioni per supportare un'esperienza di console modificata utilizzando la policy `AmazonElastiCacheFullAccess`.

27 novembre 2023

[ElastiCache \(RedisOSS\) ora supporta la modifica della modalità cluster](#)

È ora possibile migrare i cluster da Cluster Mode Disabled (CMD) a Cluster Mode Enabled (CME). Per ulteriori informazioni, consultare [Modifica della modalità cluster](#).

11 maggio 2023

[ElastiCache \(RedisOSS\) ora supporta la modifica delle impostazioni di crittografia in transito](#)

Ora puoi modificare la TLS configurazione dei cluster Redis senza dover ricostruire o rifornire OSS i 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 \(RedisOSS\) ora supporta l'autenticazione degli utenti tramite IAM](#)

IAM L'autenticazione consente di autenticare una connessione a ElastiCache (RedisOSS) utilizzando identità. AWS IAM 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 \(RedisOSS\) ora supporta Redis 7 OSS](#)

Questa versione offre diverse nuove funzionalità ad Amazon ElastiCache (RedisOSS): OSS funzioni Redis, ACL miglioramenti e Sharded Pub/Sub. [Per ulteriori informazioni, consulta ElastiCache \(Redis\) versione 7.0. OSS](#)

8 novembre 2022

## [ElastiCache \(RedisOSS\) ora supporta IPV6](#)

7 novembre 2022

ElastiCache supporta le versioni 4 e 6 (IPv4eIPv6) del protocollo Internet, che consente di configurare il cluster in modo che accetti solo IPv4 connessioni, solo connessioni o entrambe IPv4 e IPv6 IPv6 connessioni (dual-stack). IPv6 è supportato per i [carichi di lavoro che utilizzano o il OSS motore Redis dalla versione 6.2 in poi su tutte le istanze basate sul sistema Nitro](#). Non sono previsti costi aggiuntivi per l'accesso tramite. ElastiCache IPv6 Per ulteriori informazioni, consulta [Scelta di un tipo di rete](#).

## [ElastiCache \(Memcached\) ora supporta IPV6](#)

ElastiCache supporta le versioni 4 e 6 (IPv4eIPv6) del protocollo Internet, che consente di configurare il cluster in modo che accetti solo IPv4 connessioni, solo connessioni o entrambe IPv4 e IPv6 IPv6 connessioni (dual-stack). IPv6 è supportato per i carichi di lavoro che utilizzano o il motore Memcached dalla versione 1.6.6 in poi su tutte le istanze basate sul sistema Nitro. Non sono previsti costi aggiuntivi per l'accesso tramite. ElastiCache IPv6 Per ulteriori informazioni, consulta [Scelta di un tipo di rete](#).

7 novembre 2022

## [ElastiCache \(Memcached\) ora supporta la crittografia in transito](#)

La crittografia in transito è una funzione opzionale che consente di aumentare la sicurezza dei dati nei punti più vulnerabili, ossia quando sono in transito da una posizione all'altra. È supportata dalle versioni di Memcached 1.6.12 e successive. Per ulteriori informazioni, vedete la crittografia [ElastiCache in transito](#) (). TLS

26 maggio 2022

[ElastiCache \(RedisOSS\) 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. OSS È possibile archiviare e accedere in modo nativo ai dati utilizzando il formato JavaScript Object Notation (JSON) all'interno dei cluster Redis e aggiornarli e i JSON dati archiviati in tali OSS cluster, senza dover gestire codice personalizzato per serializzarli e deserializzarli. [Per ulteriori informazioni, consulta Guida introduttiva a. JSON](#)

25 maggio 2022

[ElastiCache ora supporta PrivateLink](#)

AWS PrivateLink consente di accedere in modo privato alle ElastiCache API operazioni senza un gateway Internet, un NAT dispositivo, una connessione o una VPN connessione AWS Direct Connect. Per ulteriori informazioni, consulta [Amazon ElastiCache API and interface VPC endpoints \(AWS PrivateLink\)](#) per Redis o OSS [Amazon ElastiCache API and interface VPC endpoints \(AWS PrivateLink\)](#) per Memcached.

24 gennaio 2022

[ElastiCache \(RedisOSS\) ora supporta Redis 6.2 e Data Tiering OSS](#)

Amazon ElastiCache (RedisOSS) introduce la prossima versione del OSS motore Redis supportato da Amazon. ElastiCache ElastiCache (RedisOSS) 6.2 include miglioramenti delle prestazioni per i cluster TLS abilitati che utilizzano tipi di nodi x86 con 8 vCPUs o più o tipi di nodi Graviton2 con 4 o più. vCPUs ElastiCache (Redis) introduce anche il tiering dei dati. OSS 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 \(RedisOSS\) versione 6.2 \(avanzata\) e Data tiering.](#)

23 novembre 2021

[Support per Auto Scaling](#)

ElastiCache (RedisOSS) ora supporta Auto Scaling. ElastiCache (RedisOSS) la scalabilità automatica è la capacità di aumentare o diminuire automaticamente gli shard o le repliche desiderati nel servizio ElastiCache (RedisOSS). ElastiCache sfrutta il servizio Application Auto Scaling per fornire questa funzionalità. Per ulteriori informazioni, consulta [Cluster Auto Scaling ElastiCache \(Redis OSS\)](#).

19 agosto 2021

[Support per la distribuzione di Redis OSS Slow log](#)

ElastiCache ora consente di eseguire lo streaming di Redis OSS SLOWLOG su una delle due destinazioni: Amazon Data Firehose o Amazon CloudWatch Logs. Per ulteriori informazioni, consulta [Consegna dei registri](#).

22 aprile 2021

[Support per assegnazione di tag a risorse e chiavi di condizione](#)

ElastiCache ora supporta il tagging per aiutarti a gestire i cluster e altre risorse. ElastiCache Per ulteriori informazioni, consulta [Taggare](#) le risorse. ElastiCache ElastiCache introduce anche il supporto per le chiavi di condizione. È possibile specificare le condizioni che determinano l'effetto di una IAM politica. Per ulteriori informazioni, consulta [Utilizzo delle chiavi di condizione](#).

7 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 ElastiCache risorse. Per ulteriori informazioni, consulta [Taggare](#) le risorse. ElastiCache ElastiCache introduce anche il supporto per le chiavi di condizione. È possibile specificare le condizioni che determinano l'effetto di una IAM politica. 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 [Using Outposts for Redis](#) o [OSS Using Outposts](#) for Memcached.

8 ottobre 2020

### [ElastiCache OSSora supporta Redis 6](#)

Amazon ElastiCache (RedisOSS) introduce la prossima versione del OSS 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 [ElastiCache \(RedisOSS\) Versione 6.0 \(Enhanced\)](#).

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 \(RedisOSS\) ora supporta la scalabilità dell'ambiente Redis OSS Cluster fino a 500 nodi o 500 shard](#)

La modalità Redis OSS Cluster rende possibili configurazioni utilizzabili per partizionare i dati su più shard e offre una migliore scalabilità, prestazioni e disponibilità. Questa funzionalità è disponibile su Amazon ElastiCache (RedisOSS) dalla versione 5.0.6 in poi in tutte le AWS regioni e per tutti gli ambienti cluster esistenti e nuovi ElastiCache (RedisOSS). Per ulteriori informazioni, consulta [Redis OSS Nodes and Shards](#).

13 agosto 2020

[ElastiCache ora supporta le autorizzazioni a livello di risorsa](#)

Ora puoi limitare l'ambito delle autorizzazioni di un utente ElastiCache specificando le risorse in una politica (). AWS Identity and Access Management IAM Per ulteriori informazioni, consulta [Autorizzazioni a livello di risorsa](#).

12 agosto 2020

[ElastiCache \(RedisOSS\) aggiunge parametri Amazon CloudWatch aggiuntivi](#)

ElastiCache (RedisOSS) ora supporta nuove CloudWatch metriche, tra cui e. PubSubCmds HyperLogLogBasedCmds Per un elenco completo, consulta [Metrics](#) for Redis. OSS

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 \(RedisOSS\) ora supporta Global Datastore per Redis OSS](#)

La OSS funzionalità Global Datastore for Redis offre 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 ElastiCache (RedisOSS) 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 \(RedisOSS\) ora supporta la versione Redis 5.0.6 OSS](#)

Per ulteriori informazioni, consulta [ElastiCache \(RedisOSS\) Versione 5.0.6 \(Enhanced\)](#).

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 EC2 di Amazon offrono un livello di CPU prestazioni di base con la possibilità di aumentare l'CPU utilizzo in qualsiasi momento fino all'esaurimento dei crediti accumulati. Per ulteriori informazioni, consulta [Tipi di nodi supportati](#).

12 novembre 2019

[Amazon ElastiCache ora supporta la modifica del AUTH token su un server esistente ElastiCache \(RedisOSS\)](#)

ElastiCache (RedisOSS) 5.0.6 30 ottobre 2019

ora consente di modificar 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 Redis OSS AUTH](#).

[Amazon ElastiCache ora supporta la migrazione dei dati online da Redis OSS su Amazon EC2](#)

Ora puoi utilizzare Online Migration per migrare i tuoi dati da Redis self-hosted su OSS Amazon ad EC2 Amazon. ElastiCache Per ulteriori informazioni, consulta Migrazione [online](#) a. ElastiCache

28 ottobre 2019

[ElastiCache \(RedisOSS\) introduce la scalabilità verticale online per la modalità Redis Cluster. OSS](#)

Ora puoi scalare verso l'alto o verso il basso il tuo Redis Cluster condiviso su richiesta. OSS ElastiCache (RedisOSS) 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 \(RedisOSS\) ora consente agli utenti di utilizzare un singolo endpoint di lettura per il cluster Amazon ElastiCache \(RedisOSS\).](#)

Questa funzionalità consente di indirizzare tutto il traffico di lettura verso il cluster ElastiCache (RedisOSS) 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 \(RedisOSS\) 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, tutto anticipato e nessun anticipo.](#)

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 \(RedisOSS\) supporto per un massimo di 250 nodi per cluster Redis OSS](#)

Il limite di nodi o shard può essere aumentato fino a un massimo di 250 per cluster ElastiCache (RedisOSS). Per ulteriori informazioni, consulta [Shard](#).

19 novembre 2018

[ElastiCache \(RedisOSS\) supporto per il failover automatico e il backup e il ripristino su tutti i nodi T2](#)

ElastiCache (RedisOSS) introduce il supporto per il failover automatico, la creazione di istantanee e il backup e il ripristino su tutti i nodi T2. Per ulteriori informazioni, consulta [ElastiCache \(RedisOSS\) Backup e ripristino e istantanea](#).

19 novembre 2018

[ElastiCache \(RedisOSS\) supporto per i nodi M5 e R5](#)

ElastiCache (RedisOSS) 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 (RedisOSS) 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 per l'utente ElastiCache (RedisOSS). Vedi anche [DecreaseReplicaCount](#) e [IncreaseReplicaCount](#) nella Guida di riferimento. ElastiCache API

17 settembre 2018

[Certificazione di RAMP conformità Fed](#)

ElastiCache (RedisOSS) è ora certificato per la RAMP conformità alla Fed. Per ulteriori informazioni, consulta [Convalida della conformità per Amazon ElastiCache](#).

30 agosto 2018

[Aggiornamenti del motore Valkey o Redis OSS \(modalità cluster abilitata\)](#)

Amazon ElastiCache (RedisOSS) ha aggiunto il supporto per l'aggiornamento delle versioni del motore Valkey o Redis OSS (abilitata per la modalità cluster). Per ulteriori informazioni, consulta la sezione dedicata all'[aggiornamento delle versioni di un motore](#).

20 agosto 2018

[PCIDSScertificazione di conformità](#)

ElastiCache (RedisOSS) è ora certificata per la PCI DSS conformità. Per ulteriori informazioni, consulta [Convalida della conformità per Amazon ElastiCache.](#)

5 luglio 2018

[Supporto per ElastiCache \(RedisOSS\) 4.0.10](#)

ElastiCache (RedisOSS) ora supporta Redis OSS 4.0.10, inclusa la crittografia e il ridimensionamento online del cluster in un'unica versione. Per ulteriori informazioni, consulta [ElastiCache \(Redis\) Versione 4.0.10 \(Enhanced OSS\).](#)

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 OSS \(\(Redis\) User Guide OSS\) e per Memcached ElastiCache \(\(Memcached\) 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 EngineCPU Utilization metric](#)

ElastiCache (RedisOSS) ha aggiunto una nuova metrica che riporta la percentuale della capacità CPU dell'utente attualmente utilizzata. EngineCPUUtilization Per ulteriori informazioni, consulta [Metrics](#) for RedisOSS

9 aprile 2018

La tabella seguente descrive le modifiche importanti alla Guida per l'ElastiCache utente prima di marzo 2018.

Modifica	Descrizione	Data della modifica
Support per Regione Asia Pacifico (Osaka-Locale)	<p>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:</p> <ul style="list-style-type: none"> <li>• <a href="#">Regioni supportate</a></li> <li>• <a href="#">Tipi di nodi di cache supportati</a></li> </ul>	12 febbraio 2018
Support per l'UE (Parigi).	<p>ElastiCache ha aggiunto il supporto per la regione UE (Parigi). Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"> <li>• <a href="#">Regioni supportate</a></li> <li>• <a href="#">Tipi di nodi di cache supportati</a></li> </ul>	18 dicembre 2017

Modifica	Descrizione	Data della modifica
Support per la regione Cina (Ningxia)	<p>Amazon ElastiCache ha aggiunto il supporto per la regione Cina (Ningxia). Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Regioni supportate</a></li><li>• <a href="#">Tipi di nodi di cache supportati</a></li></ul>	11 dicembre 2017
Support per i ruoli collegati ai servizi	<p>Questa versione ElastiCache ha aggiunto il supporto per Service Linked Roles (SLR). Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Utilizzo di ruoli collegati ai servizi per Amazon ElastiCache</a></li><li>• <a href="#">Configura le tue autorizzazioni (solo nuovi ElastiCache utenti)</a></li></ul>	7 dicembre 2017
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 e 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"><li>• <a href="#">Tipi di nodi di cache supportati</a></li><li>• <a href="#">Parametri specifici del tipo di nodo Memcached</a></li><li>• <a href="#">Parametri specifici del tipo di nodo Redis OSS</a></li></ul>	20 novembre 2017

Modifica	Descrizione	Data della modifica
ElastiCache (RedisOSS) 3.2.10 e supporto per il resharding online	<p>Amazon ElastiCache (RedisOSS) aggiunge il supporto per ElastiCache (RedisOSS) 3.2.10. ElastiCache (RedisOSS) introduce anche il ridimensionamento online del cluster per aggiungere e 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"><li>• <a href="#">Ridimensionamento di cluster online</a></li><li>• <a href="#">Resharding online per Valkey o Redis OSS (modalità cluster abilitata)</a></li></ul>	9 novembre 2017
HIPAAidoneità	<p>ElastiCache (RedisOSS) la versione 3.2.6 è ora certificata per l'HIPAAidoneità quando la crittografia è abilitata sul cluster. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Convalida della conformità per Amazon ElastiCache</a></li><li>• <a href="#">Sicurezza dei dati in Amazon ElastiCache</a></li></ul>	2 Novembre 2017

Modifica	Descrizione	Data della modifica
ElastiCache (RedisOSS) 3.2.6 e supporto per la crittografia	<p>ElastiCache aggiunge il supporto per ElastiCache (RedisOSS) 3.2.6, che include due funzionalità di crittografia:</p> <ul style="list-style-type: none"><li>• 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.</li><li>• La crittografia dei dati inattivi esegue la crittografia su disco dei dati durante operazioni di sincronizzazione e backup.</li></ul> <p>Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Sicurezza dei dati in Amazon ElastiCache</a></li><li>• <a href="#">Motori e versioni supportati</a></li></ul>	25 ottobre 2017
Argomento dei modelli di connessione	<p>ElastiCache la documentazione aggiunge un argomento che copre vari modelli di accesso a un ElastiCache cluster in AmazonVPC.</p> <p>Per ulteriori informazioni, consulta <a href="#">Modelli di accesso per accedere a una ElastiCache cache in Amazon VPC</a> la Guida ElastiCache per l'utente.</p>	24 Aprile 2017
Support per Memcached 1.4.34	<p>ElastiCache aggiunge il supporto alla versione 1.4.34 di Memcached, che incorpora una serie di correzioni alle versioni precedenti di Memcached.</p> <p><a href="#">Per ulteriori informazioni, vedere le note di rilascio di Memcached 1.4.34 su Memcached su GitHub</a></p>	10 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. OSS Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Test del failover automatico</a> nella Guida per l'utente di ElastiCache .</li><li>• <a href="#">TestFailover</a> nel Reference. ElastiCache API</li><li>• <a href="#">test-failover</a> nella documentazione di riferimento dell'AWS CLI .</li></ul>	4 Aprile 2017
Ripristino Redis OSS avanzato	<p>ElastiCache aggiunge OSS backup e ripristino Redis avanzati 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. (Inoltre CLI, questa funzionalità può ripristinare un numero diverso di gruppi di nodi anziché un numero diverso di shard.) API Questo aggiornamento supporta anche diverse configurazioni di OSS slot Redis. Per ulteriori informazioni, consulta <a href="#">Ripristino da un backup in una nuova cache</a>.</p>	15 marzo 2017



Modifica	Descrizione	Data della modifica
Nuovo parametro di gestione della memoria Redis OSS	<p>ElastiCache aggiunge un nuovo OSS parametro <code>Redisreserved-memory-percent</code>, che semplifica la gestione della memoria riservata. Questo parametro è disponibile in tutte le versioni di ElastiCache (RedisOSS). Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Gestione della memoria riservata per Valkey e Redis OSS</a></li><li>• <a href="#">Nuovi parametri per Redis 3.2.4 OSS</a></li></ul>	15 marzo 2017
Support per Memcached 1.4.33	<p>ElastiCache aggiunge il supporto per la versione 1.4.33 di Memcached. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Memcached versione 1.4.33</a></li><li>• <a href="#">Parametri di Memcached 1.4.33 aggiunti</a></li></ul>	20 dicembre 2016
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"><li>• <a href="#">Regioni supportate</a></li><li>• <a href="#">Tipi di nodi di cache supportati</a></li></ul>	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 nodo M4 e T2. AWS Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Regioni supportate</a></li><li>• <a href="#">Tipi di nodi di cache supportati</a></li></ul>	8 dicembre 2016

Modifica	Descrizione	Data della modifica
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"><li>• <a href="#">Regioni supportate</a></li><li>• <a href="#">Tipi di nodi di cache supportati</a></li></ul>	1° novembre 2016
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 tipi di nodi M4, T2 e R3. Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"><li>• <a href="#">Regioni supportate</a></li><li>• <a href="#">Tipi di nodi di cache supportati</a></li></ul>	17 ottobre 2016

Modifica	Descrizione	Data della modifica
Support per Redis Cluster OSS	<p>ElastiCache aggiunge il supporto per Redis OSS Cluster (avanzato). I clienti che utilizzano Redis OSS Cluster possono partizionare i propri dati su un massimo di 15 shard (gruppi di nodi). Ognle partizioni supporta la replica (fino a cinque repliche di lettura per partizione). I tempi di failover automatico di Redis OSS Cluster sono circa un quarto rispetto a quelli delle 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"><li>• <a href="#">Confronto tra Memcached e Redis OSS</a></li><li>• <a href="#">ElastiCache componenti e caratteristiche:</a> annotare le sezioni su Nodi, Frammenti, Cluster e Replica.</li><li>• <a href="#">ElastiCache terminologia</a></li></ul>	12 Ottobre 2016
Support per i tipi di nodi M4	<p>ElastiCache aggiunge il supporto per la famiglia di tipi di nodi M4 nella maggior parte AWS delle 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:</p> <ul style="list-style-type: none"><li>• <a href="#">Tipi di nodi di cache supportati</a></li><li>• <a href="#">Parametri specifici del tipo di nodo Memcached</a></li><li>• <a href="#">Parametri specifici del tipo di nodo Redis OSS</a></li></ul>	3 agosto 2016

Modifica	Descrizione	Data della modifica
Support per la regione di Mumbai	<p>ElastiCache aggiunge il supporto per la regione Asia Pacifico (Mumbai). Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"> <li>• <a href="#">Tipi di nodi di cache supportati</a></li> <li>• <a href="#">Parametri specifici del tipo di nodo Memcached</a></li> <li>• <a href="#">Parametri specifici del tipo di nodo Redis OSS</a></li> </ul>	27 giugno 2016
Esportazione di snapshot	<p>ElastiCache aggiunge la possibilità di esportare un'OSSistantanea Redis in modo da potervi accedere dall'esterno. ElastiCache Per ulteriori informazioni, consulta gli argomenti seguenti:</p> <ul style="list-style-type: none"> <li>• <a href="#">Esportazione di un backup</a> nella Amazon ElastiCache User Guide</li> <li>• <a href="#">CopySnapshot</a> nel Amazon ElastiCache API Reference</li> </ul>	26 maggio 2016
Dimensionamento verso l'alto del tipo di nodo	<p>ElastiCache aggiunge la possibilità di scalare il tipo di OSS nodo Redis. Per ulteriori informazioni, consulta <a href="#">Ridimensionamento ElastiCache</a>.</p>	24 marzo 2016
Aggiornamento del motore semplificato	<p>ElastiCache aggiunge la possibilità di aggiornare e facilmente il motore di OSS cache Redis. Per ulteriori informazioni, consulta <a href="#">Gestione delle versioni per ElastiCache</a>.</p>	22 marzo 2016
Support per i tipi di nodi R3	<p>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 <a href="#">Tipi di nodi di cache supportati</a>.</p>	16 marzo 2016

Modifica	Descrizione	Data della modifica
Accesso ElastiCache tramite una funzione Lambda	È stato aggiunto un tutorial sulla configurazione di una funzione Lambda per ElastiCache l'accesso in Amazon VPC. Per ulteriori informazioni, consulta <a href="#">Altri ElastiCache tutorial e video</a> .	12 febbraio 2016
Support per Redis 2.8.24 OSS	ElastiCache aggiunge il supporto per la OSS versione 2.8.24 di Redis con miglioramenti aggiunti rispetto a Redis 2.8.23. OSS I miglioramenti includono la correzione di alcuni bug e il support per la registrazione di indirizzi di accesso alla memoria con errori. Per ulteriori informazioni, consulta gli argomenti seguenti: <ul style="list-style-type: none"><li>• <a href="#">ElastiCache (RedisOSS) versione 2.8.24 (migliorata)</a></li><li>• <a href="#">Note di rilascio di Redis 2.8 OSS</a></li></ul>	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 OSS versioni più recenti di Redis offrono un'esperienza utente migliore e più stabile, OSS 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 <a href="#">Motori e versioni supportati</a> .	15 dicembre 2015

Modifica	Descrizione	Data della modifica
Supporto per Redis OSS 2.8.23.	ElastiCache aggiunge il supporto per la OSS versione Redis 2.8.23 con miglioramenti aggiunti rispetto a Redis 2.8.22. OSS I miglioramenti includono la correzione di alcuni bug e il support 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 <a href="#">ElastiCache (RedisOSS) versione 2.8.23 (migliorata)</a> .	13 Novembre 2015

Modifica	Descrizione	Data della modifica
Support per Redis OSS 2.8.22.	<p>ElastiCache aggiunge il supporto per la OSS versione Redis 2.8.22 con miglioramenti e miglioramenti ElastiCache aggiunti rispetto alla versione 2.8.21. Miglioramenti:</p> <ul style="list-style-type: none"><li data-bbox="402 457 1097 659">• Implementazione di un processo di salvataggio senza fork che consente di salvare quando la scarsa memoria disponibile rischia di bloccare un salvataggio con fork.</li><li data-bbox="402 695 1019 793">• SaveInProgressReplicationBytesMetriche aggiuntive e. CloudWatch</li><li data-bbox="402 829 1110 978">• Per abilitare le sincronizzazioni parziali, il OSS parametro Redis <code>repl-backlog-size</code> ora si applica a tutti i cluster.</li></ul> <p>Per l'elenco completo delle modifiche e ulteriori informazioni, consulta <a href="#">ElastiCache (RedisOSS) versione 2.8.22 (migliorata)</a>.</p> <p>Questa versione di documentazione include una riorganizzazione della documentazione e la rimozione della documentazione dell'interfaccia a riga di ElastiCache comando (). CLI Per l'utilizzo da riga di comando, fare riferimento alla <a href="#">AWS riga di comando</a> per ElastiCache.</p>	28 settembre 2015

Modifica	Descrizione	Data della modifica
Support per Memcached 1.4.28.	ElastiCache aggiunge il supporto per la versione 1.4.24 di Memcached e i miglioramenti di Memcached dalla versione 1.4.14. Questa versione aggiunge il supporto per la gestione della cache utilizzata meno di recente (LRU) come attività in background, la scelta di jenkins o murmur3 come algoritmo di hashing, nuovi comandi e varie correzioni di bug. Per ulteriori informazioni, consulta le <a href="#">note di rilascio di Memcached</a> .	27 agosto 2015
Support per Memcached Auto Discovery con 5.6 PHP	Questa versione di Amazon ElastiCache aggiunge il supporto per il client Memcached Auto Discovery per la PHP versione 5.6. Per ulteriori informazioni, consulta <a href="#">Compilazione del codice sorgente per il client del ElastiCache cluster per PHP</a> .	29 luglio 2015
Supporto per Redis 2.8.21 OSS	ElastiCache aggiunge il supporto per la OSS versione 2.8.21 di Redis e i miglioramenti di Redis rispetto alla versione 2.8.19. OSS Questa versione di Redis OSS include diverse correzioni di bug. Per ulteriori informazioni, consulta le note di rilascio di <a href="#">Redis OSS 2.8</a> .	29 luglio 2015
Nuovo argomento : Accesso ElastiCache dall'esterno AWS	È stato aggiunto un nuovo argomento su come accedere alle ElastiCache risorse dall'esterno AWS. Per ulteriori informazioni, vedere <a href="#">Accesso ElastiCache dall'esterno AWS</a> .	9 luglio 2015



Modifica	Descrizione	Data della modifica
Aggiunta di messaggi relativi alla sostituzione di nodi	<p>ElastiCache aggiunge tre messaggi relativi alla sostituzione pianificata dei nodi, ElastiCache: NodeReplacementScheduled, ElastiCache: NodeReplacementRescheduled, ElastiCache: NodeReplacementCanceled.</p> <p>Per ulteriori informazioni e azioni che puoi intraprendere quando è pianificata la sostituzione di un nodo, ElastiCache consulta <a href="#">Notifiche di eventi e Amazon SNS</a>.</p>	11 giugno 2015

Modifica	Descrizione	Data della modifica
<p>Support per Redis OSS v. 2.8.19.</p>	<p>ElastiCache aggiunge il supporto per la OSS versione 2.8.19 di Redis e i miglioramenti di Redis rispetto alla versione 2.8.6. OSS Support garantito per:</p> <ul style="list-style-type: none"><li>• La struttura HyperLogLog dei dati, con i comandi Redis, e. OSS PFADD PFCOUNT PFMERGE</li><li>• Interrogazioni sull'intervallo lessicografico con i nuovi comandi ZRANGEBYLEX, e. ZLEXCOUNT ZREMRANGEBYLEX</li><li>• Sono state introdotte diverse correzioni di bug, in particolare impedendo a un nodo primario di inviare dati obsoleti ai nodi di replica bloccando il primario SYNC quando un processo secondario di salvataggio in background (bgsave) termina in modo imprevisto.</li></ul> <p><a href="#">Per ulteriori informazioni su HyperLogLog, consulta la nuova struttura dati di Redis: the. OSS HyperLogLog</a></p> <p>Per ulteriori informazioni su PFADD, e PFCOUNTPFMERGE, consulta la <a href="#">OSS documentazione Redis</a> e fai clic su. HyperLogLog</p>	<p>11 marzo 2015</p>
<p>Support per i tag di allocazione dei costi</p>	<p>ElastiCache aggiunge il supporto per i tag di allocazione dei costi. Per ulteriori informazioni, consulta <a href="#">Monitoraggio dei costi con i tag di allocazione dei costi</a>.</p>	<p>9 febbraio 2015</p>

Modifica	Descrizione	Data della modifica
Supporto per la AWS GovCloud regione (Stati Uniti occidentali)	ElastiCache aggiunge il supporto per la regione AWS GovCloud (Stati Uniti occidentali) (us-gov-west-1).	29 gennaio 2015
Supporto per la regione Europa (Francoforte)	ElastiCache aggiunge il supporto per la regione Europa (Francoforte) (eu-central-1).	19 gennaio 2015
Supporto Multi-AZ per i gruppi di replica Redis OSS	ElastiCache aggiunge il supporto per Multi-AZ dal nodo primario a una replica di lettura in un gruppo di replica Redis. OSS 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 <a href="#">Riduzione al minimo dei tempi di inattività a ElastiCache utilizzando Multi-AZ con Valkey e Redis OSS</a> .	24 Ottobre 2014
AWS CloudTrail I registrazione delle chiamate supportata API	ElastiCache aggiunge il supporto per l'utilizzo AWS CloudTrail per registrare tutte le ElastiCache API chiamate. Per ulteriori informazioni, consulta <a href="#">Registrazione delle ElastiCache API chiamate Amazon con AWS CloudTrail</a> .	15 settembre 2014
Nuove dimension i d'istanza supportate	ElastiCache aggiunge il supporto per ulteriori istanze General Purpose (T2). Per ulteriori informazioni, consulta <a href="#">Configurazione dei parametri del motore utilizzando gruppi di ElastiCache parametri</a> .	11 settembre 2014
Supportato il posizionamento di nodi flessibile per Memcached	ElastiCache aggiunge il supporto per la creazione di nodi Memcached su più zone di disponibilità.	23 luglio 2014

Modifica	Descrizione	Data della modifica
Nuove dimensioni d'istanza supportate	ElastiCache aggiunge il supporto per ulteriori istanze General Purpose (M3) e istanze Memory Optimized (R3). Per ulteriori informazioni, consulta <a href="#">Configurazione dei parametri del motore utilizzando gruppi di ElastiCache parametri</a> .	1 luglio 2014
PHPindividuazione automatica	È stato aggiunto il supporto per l'individuazione automatica della PHP versione 5.5.	13 maggio 2014
Backup e ripristino per cluster Redis OSS	In questa versione, ElastiCache consente ai clienti di creare istantanee dei propri cluster Redis e creare nuovi OSS cluster utilizzando queste istantanee. Un backup è una copia del cluster in un momento specifico ed è costituito dai metadati del cluster e da tutti i dati nella cache Redis. OSS 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 <a href="#">Snapshot e ripristino</a> .	24 Aprile 2014
Redis 2.8.6 OSS	ElastiCache supporta Redis OSS 2.8.6, oltre a Redis 2.6.13. OSS Con Redis OSS 2.8.6, i clienti possono migliorare la resilienza e la tolleranza agli errori delle repliche di lettura, con il supporto per la risincronizzazione parziale e un numero minimo definito dall'utente di repliche di lettura che devono essere sempre disponibili. Redis OSS 2.8.6 offre anche il supporto completo per, in cui i client possono essere informati degli eventi che si verificano sul publish-and-subscribe server.	13 marzo 2014

Modifica	Descrizione	Data della modifica
Motore di cache Redis OSS	<p>ElastiCache offre il software Redis OSS Cache Engine, oltre a Memcached. I clienti che attualmente utilizzano Redis OSS possono «seminare» un nuovo cluster di OSS cache ElastiCache Redis con i dati esistenti da un file di OSS snapshot Redis, facilitando la migrazione verso un ambiente gestito. ElastiCache</p> <p>Per supportare le funzionalità di replica di Redis, ora supporta i OSS gruppi di replica. ElastiCache API I clienti possono creare un gruppo di replica con un nodo di OSS cache Redis primario e aggiungere uno o più nodi di replica di lettura che rimangono 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
Supporto per Amazon Virtual Private Cloud predefinito (VPC)	<p>In questa versione, ElastiCache è completamente integrato con Amazon Virtual Private Cloud (VPC). Per i nuovi clienti, i cluster di cache vengono creati in Amazon VPC per impostazione predefinita. Per ulteriori informazioni, consulta <a href="#">Amazon VPCs e la ElastiCache sicurezza</a>.</p>	8 gennaio 2013
PHPsupporto per l'individuazione automatica dei nodi di cache	<p>L'individuazione automatica dei nodi di cache, nella versione iniziale, offre support per i programmi Java. In questa versione, ElastiCache porta il supporto per l'individuazione automatica dei nodi di cache aPHP.</p>	2 gennaio 2013

Modifica	Descrizione	Data della modifica
Supporto 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 VPC automaticamente in Amazon; i clienti esistenti possono migrare ad Amazon VPC al proprio ritmo. Per ulteriori informazioni, consulta <a href="#">Amazon VPCs e la ElastiCache sicurezza</a> .	20 dicembre 2012
Individuazione automatica dei nodi di cache e nuova versione del motore della cache	<p>ElastiCache fornisce l'individuazione automatica dei nodi di cache, ovvero la possibilità per i programmi client di determinare automaticamente tutti i nodi di cache in un cluster e di avviare e mantenere le connessioni a tutti questi nodi.</p> <p>Questa release offre, inoltre, una nuova versione del motore della cache: Memcached 1.4.14. Questo motore della cache inedito vanta, rispetto al passato, caratteristiche ottimizzate di ribilanciamento dell'allocazione, nonché notevoli miglioramenti in termini di prestazioni e scalabilità e varie correzioni di bug. Mette a disposizione, inoltre, molti nuovi parametri della cache configurabili. Per ulteriori informazioni, consulta <a href="#">Configurazione dei parametri del motore utilizzando gruppi di ElastiCache parametri</a>.</p>	28 Novembre 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 supporto 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.