



Leitfaden

Amazon ElastiCache



API-Version 2015-02-02

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

Amazon ElastiCache: Leitfaden

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist ElastiCache?	1
Serverless-Caching	1
Selbst entworfene Cluster	2
Zugehörige Services	2
Funktionsweise	3
Cache- und Caching-Engines	4
Auswählen aus den Bereitstellungsoptionen	10
ElastiCache Ressourcen	17
AWS Regionen und Verfügbarkeitszonen	19
Anwendungsfälle	21
Erste Schritte mit ElastiCache	30
einrichten ElastiCache	30
Melden Sie sich an für ein AWS-Konto	30
Erstellen eines Benutzers mit Administratorzugriff	31
Erteilen programmgesteuerten Zugriffs	32
Berechtigungen einrichten	34
Einrichten EC2	35
Gewähren des Netzwerkzugriffs	36
Richten Sie den Befehlszeilenzugriff ein	37
Erstellen Sie einen serverlosen Valkey-Cache	38
Lesen und Schreiben von Daten	39
Bereinigen	41
Nächste Schritte	42
Erstellen Sie einen serverlosen Valkey- oder Redis-Cache OSS	42
Lesen und Schreiben von Daten	44
Bereinigen	46
Nächste Schritte	47
Erstellen Sie einen serverlosen Memcached-Cache	47
Lesen und Schreiben von Daten	48
Bereinigen	53
Nächste Schritte	54
Tutorials: Erste Schritte mit Python und ElastiCache	54
Python und ElastiCache	55
Tutorial: Konfiguration von Lambda für den Zugriff ElastiCache in einem VPC	73

Schritt 1: Einen serverlosen Cache erstellen ElastiCache .	73
Schritt 2: Erstellen Sie eine Lambda-Funktion für ElastiCache	76
Schritt 3: Testen Sie die Lambda-Funktion mit ElastiCache	80
Schritt 4: Aufräumen (optional)	81
Entwerfen Sie Ihren eigenen ElastiCache Cluster	83
Komponenten und Funktionen	83
Knoten	84
ElastiCache Scherben	85
ElastiCache Cluster	85
ElastiCache Replikation	88
ElastiCache Endpunkte	90
Parametergruppen	91
ElastiCache Sicherheit	92
Subnetzgruppen	92
ElastiCache Backups	92
Ereignisse	93
ElastiCache Terminologie	94
Tutorial: So entwerfen Sie Ihren eigenen Cluster	96
Entwerfen Sie Ihren eigenen ElastiCache (Valkey-) Cluster	96
Entwerfen Sie Ihren eigenen ElastiCache (Redis-OSS) Cluster	119
Löschen eines Clusters	141
Weitere Tutorials und Videos	143
Videos	144
Knoten verwalten in ElastiCache	149
ElastiCache Knotenstatus anzeigen	150
Valkey- oder OSS Redis-Knoten und -Shards	156
Mit Knoten verbinden	158
Unterstützte Knotentypen	163
Knoten neu starten	178
Knoten ersetzen (Valkey und OSS Redis)	183
Knoten ersetzen (Memcached)	191
Reservierte Knoten	193
Migration von Knoten der vorherigen Generation	209
Verwaltung von Clustern in ElastiCache	213
Auswahl eines Netzwerktyps in ElastiCache	215
Automatische Erkennung (Memcached)	220

Daten-Tiering ElastiCache	264
Einen Cluster vorbereiten in ElastiCache	271
Einen Cluster für Valkey oder Redis erstellen OSS	283
Einen Cluster für Memcached erstellen	293
Details eines ElastiCache Clusters anzeigen	297
Einen ElastiCache Cluster ändern	311
Hinzufügen von Knoten zu einem ElastiCache Cluster	317
Knoten aus einem ElastiCache Cluster entfernen	328
Abbrechen ausstehender Operationen zum Hinzufügen oder Löschen von Knoten in ElastiCache	337
Löschen eines Clusters in ElastiCache	338
Zugreifen auf Ihren ElastiCache Cluster oder Ihre Replikationsgruppe	341
Verbindungsendpunkte finden in ElastiCache	350
Scherben rein ElastiCache	366
Vergleich der selbst entworfenen Caches von ValkeyOSS, Redis und Memcached	372
Online-Migration für Valkey oder Redis OSS	378
Übersicht	378
Schritte zur Migration	379
Vorbereiten Ihrer Quelle und Ihres Ziels für die Migration	379
Testen der Datenmigration	381
Migration starten	381
Verifizieren des Fortschritts der Datenmigration	383
Datenmigration abschließen	384
Durchführen einer Online-Datenmigration mithilfe der Konsole	385
Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache	386
Überlegungen zur Availability Zone bei Memcached	387
Lokalisieren Ihrer Knoten	390
Unterstützte Regionen und Endpunkte	390
Verwenden von lokalen Zonen mit ElastiCache	395
Outposts verwenden mit ElastiCache	397
Arbeiten mit ElastiCache	402
Snapshot und Wiederherstellung	402
Beschränkungen	403
Auswirkungen von Backups selbst entworfener Cluster auf die Leistung	404
Planen automatischer Backups	406
Erstellen manueller Backups	408

Erstellen einer endgültigen Sicherung	414
Beschreiben von Sicherungen	417
Kopieren eines Backups	419
Exportieren einer Sicherung	421
Wiederherstellen aus einem Backup	429
Löschen einer Sicherung	432
Markieren von Sicherungen	433
Tutorial: Seeding eines selbst entworfenen Clusters mit einem Backup	435
Motorversionen und Aufrüstung in ElastiCache	445
Versionsverwaltung für ElastiCache	445
So führen Sie ein Upgrade von Engine-Versionen aus	450
So lösen Sie ein Cross-Engine-Upgrade von Redis auf Valkey OSS aus	451
Unterstützte Versionen	452
Wesentliche Unterschiede im Versionsverhalten und in der Kompatibilität mit Valkey	475
Wesentliche Unterschiede im Versionsverhalten und in der Kompatibilität mit Redis OSS	475
Behebung blockierter Engine-Upgrades mit Valkey oder Redis OSS	479
Bewährte Methoden und Caching-Strategien	480
Allgemeine Best Practices	480
Unterstützte und eingeschränkte Valkey-, Redis OSS - und Memcached-Befehle	482
OSSKonfiguration und Limits von Valkey und Redis	519
IPv6Kundenbeispiele für Valkey, Redis OSS und Memcached	523
Best Practices für Kunden (Valkey und OSS Redis)	524
Bewährte Methoden für Kunden (Memcached)	548
TLSaktivierte ElastiCache Dual-Stack-Cluster	552
Verwaltung des reservierten Speichers für Valkey und Redis OSS	555
Bewährte Methoden bei der Arbeit mit selbst entworfenen Clustern von Valkey und Redis OSS	562
Caching-Strategien für Memcached	568
Verwaltung Ihres selbst entworfenen Clusters in ElastiCache	573
Auto Scaling Valkey- und Redis-Cluster OSS	574
Ändern des Clustermodus	623
AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher	626
Hohe Verfügbarkeit mit Replikationsgruppen	656
Verwaltung der ElastiCache Cluster-Wartung	748
Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen	750
Skalierung ElastiCache	862

ElastiCache Serverlose Skalierung	862
Festlegen von Skalierungslimits zur Kostenkontrolle	862
Vorskalierung mit Serverless ElastiCache	862
Skalierungsgrenzen mithilfe der Konsole festlegen und AWS CLI	864
Skalierung selbst entworfener Cluster	866
Erste Schritte mit JSON für Valkey und Redis OSS	945
JSONÜberblick über Datentypen	946
JSONBefehle	958
Verschlagworten Sie Ihre Ressourcen ElastiCache	1000
Überwachung von Kosten mit Tags	1012
Verwaltung von Stichwörtern mithilfe der AWS CLI	1014
Verwaltung von Stichwörtern mithilfe der ElastiCache API	1019
Amazon ElastiCache Well-Architected Lens	1022
Säule „Betriebliche Exzellenz“	1023
Säule der Sicherheit	1032
Säule der Zuverlässigkeit	1039
Säule der Leistungseffizienz	1046
Säule der Kostenoptimierung	1058
Problembekämpfung in ElastiCache	1065
Verbindungsprobleme	1065
Valkey- oder Redis-Client-Fehler OSS	1066
Fehlerbehebung bei hoher Latenz in Serverless ElastiCache	1067
Behebung von Drosselungsproblemen in Serverless ElastiCache	1069
Anhaltende Verbindungsprobleme	1070
Verwandte Themen	1092
Sicherheit	1093
Datenschutz	1094
Datensicherheit bei Amazon ElastiCache	1095
Richtlinie für den Datenverkehr zwischen Netzwerken	1177
Amazon VPCs und ElastiCache Sicherheit	1178
ElastiCache API und VPC Schnittstellen-Endpunkte (AWS PrivateLink)	1205
Subnetze und Subnetzgruppen	1208
Identitäts- und Zugriffsverwaltung	1217
Zielgruppe	1217
Authentifizierung mit Identitäten	1218
Verwalten des Zugriffs mit Richtlinien	1222

So ElastiCache arbeitet Amazon mit IAM	1225
Beispiele für identitätsbasierte Richtlinien	1232
Fehlerbehebung	1235
Zugriffskontrolle	1237
Übersicht über die Verwaltung von Zugriffsberechtigungen	1238
Compliance-Validierung	1287
Weitere Informationen	1289
Ausfallsicherheit	1289
Minimieren von Ausfällen	1290
Sicherheit der Infrastruktur	1295
Service-Updates	1295
Verwaltung von Service-Updates	1296
Sicherheitslücken wurden behoben	1301
Protokollierung und Überwachung	1304
Serverlose Metriken und Ereignisse für Valkey und Redis OSS	1304
Serverless-Metriken	1304
Serverless-Ereignisse	1315
Selbst entworfene Cluster-Metriken und Ereignisse	1332
Serverlose Metriken und Ereignisse für Memcached	1341
Serverless-Metriken	1341
Serverless-Ereignisse	1345
ElastiCache API Amazon-Anrufe protokollieren mit AWS CloudTrail	1358
ElastiCache Amazon-Informationen in CloudTrail	1358
ElastiCache Amazon-Protokolldateieinträge verstehen	1359
SNS Amazon-Ereignisüberwachung	1363
ElastiCache SNS Amazon-Benachrichtigungen verwalten	1363
ElastiCache Ereignisse anzeigen	1368
Event-Benachrichtigungen und Amazon SNS	1372
Protokollzustellung	1381
Inhalt eines langsamen Protokolleintrags	1382
Inhalte eines Engine-Protokolleintrags	1382
Berechtigungen zum Konfigurieren der Protokollierung	1382
Spezifikationen des Protokolltyps und des Protokollformats	1383
ElastiCache Ziele protokollieren	1384
Angaben der Protokollzustellung mithilfe der Konsole	1387
Angaben der Protokollzustellung mit dem AWS CLI	1388

Überwachung der Nutzung	1394
Metriken auf Host-Ebene	1394
Metriken für Valkey und Redis OSS	1398
Metriken für Memcached	1417
Welche Metriken sollte ich überwachen?	1423
Auswählen von Metrikstatistiken und -zeiträumen	1428
Überwachung von CloudWatch Cluster- und Knotenmetriken	1428
Kontingente	1432
Referenz	1434
Mit dem ElastiCache API	1434
Die Abfrage verwenden API	1434
Verfügbare Bibliotheken	1438
Fehlerbehebung bei Anwendungen	1438
Richten Sie das AWS CLI für ein ElastiCache	1440
Voraussetzungen	1440
Aufrufen der Befehlszeilen-Tools	1442
Einrichten der Tools	1442
Bereitstellung von Anmeldeinformationen für die Tools	1443
Umgebungsvariablen	1444
Fehlermeldungen	1445
Benachrichtigungen	1447
Allgemeine ElastiCache Benachrichtigungen	1447
ElastiCache (Memcached) Benachrichtigungen	1448
ElastiCache (RedisOSS) spezifische Benachrichtigungen	1448
ElastiCache Geschichte der Dokumentation	1449
AWS Glossar	1491
.....	mcdxcii

Was ist Amazon ElastiCache?

Willkommen im ElastiCache Amazon-Benutzerhandbuch. Amazon ElastiCache ist ein Webservice, der es einfach macht, einen verteilten In-Memory-Datenspeicher oder eine Cache-Umgebung in der Cloud einzurichten, zu verwalten und zu skalieren. Es handelt sich um eine leistungsstarke, skalierbare und kostengünstige Caching-Lösung. Gleichzeitig trägt es dazu bei, die mit der Bereitstellung und Verwaltung einer verteilten Cache-Umgebung verbundene Komplexität zu beseitigen.

Sie können Amazon ElastiCache in zwei Formaten betreiben. Sie können mit einem Serverless-Cache beginnen oder einen eigenen Cache-Cluster entwerfen.

Note

Amazon ElastiCache arbeitet mit den Engines Valkey, Redis und OSS Memcached. Wenn Sie sich nicht sicher sind, welche Engine Sie verwenden möchten, beachten Sie [Vergleich der selbst entworfenen Caches von ValkeyOSS, Redis und Memcached](#) in diesem Handbuch.

Serverless-Caching

ElastiCache bietet serverloses Caching, was das Hinzufügen und Betreiben eines Caches für Ihre Anwendung vereinfacht. ElastiCache Serverless ermöglicht es Ihnen, in weniger als einer Minute einen hochverfügbaren Cache zu erstellen, und macht die Bereitstellung von Instanzen oder die Konfiguration von Knoten oder Clustern überflüssig. Entwickler können einen serverlosen Cache erstellen, indem sie den Cache-Namen in der ElastiCache Konsole angeben, SDK oder CLI

ElastiCache Serverless macht außerdem die Planung und Verwaltung der Caching-Kapazität überflüssig. ElastiCache überwacht ständig den Arbeitsspeicher, die Rechenleistung und die Netzwerkbandbreite des Caches, die von Ihrer Anwendung verwendet werden, und skaliert, um den Anforderungen Ihrer Anwendung gerecht zu werden. ElastiCache bietet Entwicklern eine einfache Endpunkterfahrung, indem die zugrunde liegende Cache-Infrastruktur und das Cluster-Design abstrahiert werden. ElastiCache verwaltet die Hardwarebereitstellung, die Überwachung, den Austausch von Knoten und das Patchen von Software automatisch und transparent, sodass Sie sich auf die Anwendungsentwicklung konzentrieren können, anstatt den Cache zu betreiben.

ElastiCache Serverless ist mit Valkey 7.2, Redis OSS 7.1 und höher sowie Memcached 1.6.21 und höher kompatibel.

ElastiCache Entwerfen Sie Ihren eigenen Cluster

Wenn Sie eine genaue Kontrolle über Ihren ElastiCache Cluster benötigen, können Sie wählen, ob Sie damit Ihren eigenen Valkey-, Redis OSS - oder Memcached-Cluster entwerfen möchten. ElastiCache ermöglicht es Ihnen, Ihren Cluster zu entwerfen, indem Sie den Knotentyp, die Anzahl der Knoten und die Platzierung der Knoten in den Availability Zones für Ihren Cluster auswählen. AWS Da es ElastiCache sich um einen vollständig verwalteten Service handelt, verwaltet er automatisch die Hardwarebereitstellung, die Überwachung, den Austausch von Knoten und das Software-Patching für Ihren Cluster.

Das Entwerfen Ihres eigenen ElastiCache Clusters bietet mehr Flexibilität und Kontrolle über Ihre Cluster. Sie können beispielsweise je nach Ihren Anforderungen wählen, ob Sie einen Cluster mit Single-AZ- oder Multi-AZ-Verfügbarkeit betreiben möchten. Sie können auch wählen, ob Sie Valkey, Redis OSS oder Memcached im Clustermodus ausführen möchten, um horizontale Skalierung zu ermöglichen, oder ohne Clustermodus, um nur vertikal zu skalieren. Beim Entwerfen eines eigenen Clusters sind Sie dafür verantwortlich, den Typ und die Anzahl der Knoten richtig auszuwählen, um sicherzustellen, dass Ihr Cache über die für Ihre Anwendung erforderliche Kapazität verfügt. Sie können auch wählen, wann neue Software-Patches auf Ihren Valkey- oder Redis-Cluster angewendet werden sollen. OSS

Wenn Sie Ihren eigenen ElastiCache Cluster entwerfen, können Sie wählen, ob Sie Valkey 7.2, Redis OSS 4.0 bis 7.1 oder Memcached 1.4 und höher ausführen möchten.

Zugehörige Services

[Speicher-DB](#)

Bei der Entscheidung, ob Sie MemoryDB ElastiCache oder MemoryDB verwenden möchten, sollten Sie die folgenden Vergleiche berücksichtigen:

- ElastiCache ist ein Dienst, der häufig verwendet wird, um Daten aus anderen Datenbanken und Datenspeichern mithilfe von Valkey, OSS Redis oder Memcached zwischenspeichern. Sie sollten beim Zwischenspeichern von Workloads in Betracht ziehen ElastiCache , bei denen Sie den Datenzugriff mit Ihrer vorhandenen Primärdatenbank oder Ihrem vorhandenen Datenspeicher beschleunigen möchten (Lese- und Schreibleistung im Mikrosekundenbereich). Sie sollten auch Anwendungsfälle in Betracht ziehen ElastiCache , in denen Sie Valkey- oder OSS Redis-Datenstrukturen verwenden und auf Daten zugreifen APIs möchten, die in einer Primärdatenbank oder einem Datenspeicher gespeichert sind.

- ElastiCache kann Ihnen auch dabei helfen, Datenbankkosten zu sparen, indem häufig aufgerufene Daten in einem Cache gespeichert werden. Wenn Ihre Anwendung hohe Anforderungen an den Lesedurchsatz stellt, können Sie eine hohe Skalierbarkeit, hohe Leistung und geringere Datenspeicherkosten erzielen, indem Sie Ihre zugrunde liegende Datenbank verwenden ElastiCache, anstatt sie zu skalieren.
- MemoryDB ist eine robuste In-Memory-Datenbank für Workloads, die eine ultraschnelle Primärdatenbank erfordern. Sie ist mit Valkey und Redis kompatibel. OSS Sie sollten die Verwendung von MemoryDB in Betracht ziehen, wenn Ihr Workload eine dauerhafte Datenbank erfordert, die eine ultraschnelle Leistung bietet (Leselatenz im Mikrosekundenbereich und Schreiblatenz im einstelligen Millisekundenbereich). MemoryDB eignet sich möglicherweise auch gut für Ihren Anwendungsfall, wenn Sie eine Anwendung mit Valkey- oder OSS Redis-Datenstrukturen und APIs mit einer primären, dauerhaften Datenbank erstellen möchten. Schließlich sollten Sie erwägen, MemoryDB zu verwenden, um Ihre Anwendungsarchitektur zu vereinfachen und die Kosten zu senken, indem Sie die Verwendung einer Datenbank durch einen Cache ersetzen, um Dauerhaftigkeit und Leistung zu gewährleisten.

[Amazon Relational Database Service](#)

ElastiCache kann Ihnen helfen, Datenbankkosten zu sparen, indem Daten, auf die häufig zugegriffen wird, in einem Cache gespeichert werden. Wenn Ihre Anwendung hohe Anforderungen an den Lesedurchsatz stellt, können Sie eine hohe Skalierbarkeit, hohe Leistung und geringere Datenspeicherkosten erzielen, indem Sie Ihre zugrunde liegende Datenbank verwenden ElastiCache, anstatt sie zu skalieren.

[Weitere Hintergrundinformationen zum entsprechenden Service Amazon Relational Database Service finden Sie unter Amazon RDS](#)

ElastiCache kann Ihnen helfen, Datenbankkosten zu sparen, indem häufig aufgerufene Daten in einem Cache gespeichert werden. Wenn Ihre Anwendung hohe Anforderungen an den Lesedurchsatz stellt, können Sie eine hohe Skalierbarkeit, hohe Leistung und geringere Datenspeicherkosten erzielen, indem Sie Ihre zugrunde liegende Datenbank verwenden ElastiCache, anstatt sie zu skalieren.

Wie ElastiCache funktioniert

Hier finden Sie einen Überblick über die wichtigsten Komponenten eines ElastiCache Deployments.

Cache- und Caching-Engines

Ein Cache ist ein In-Memory-Datenspeicher, den Sie zum Speichern zwischengespeicherter Daten verwenden können. In der Regel speichert Ihre Anwendung häufig aufgerufene Daten in einem Cache, um die Antwortzeiten zu optimieren. ElastiCache bietet zwei Bereitstellungsoptionen: Serverlose und selbst entworfene Cluster. Siehe [Auswählen aus den Bereitstellungsoptionen](#).

Note

Amazon ElastiCache arbeitet mit den Engines Valkey, Redis OSS und Memcached. Wenn Sie sich nicht sicher sind, welche Engine Sie verwenden möchten, beachten Sie [Vergleich der selbst entworfenen Caches von ValkeyOSS, Redis und Memcached](#) in diesem Handbuch.

Themen

- [Wie ElastiCache funktioniert](#)
- [Preisdimensionen](#)
- [ElastiCache Backups](#)

Wie ElastiCache funktioniert

ElastiCache Serverlos

ElastiCache Serverless ermöglicht es Ihnen, einen Cache zu erstellen, ohne sich Gedanken über Kapazitätsplanung, Hardwaremanagement oder Clusterdesign machen zu müssen. Sie geben einfach einen Namen für Ihren Cache an und Sie erhalten einen einzigen Endpunkt, den Sie in Ihrem Valkey-, Redis OSS - oder Memcached-Client konfigurieren können, um mit dem Zugriff auf Ihren Cache zu beginnen.

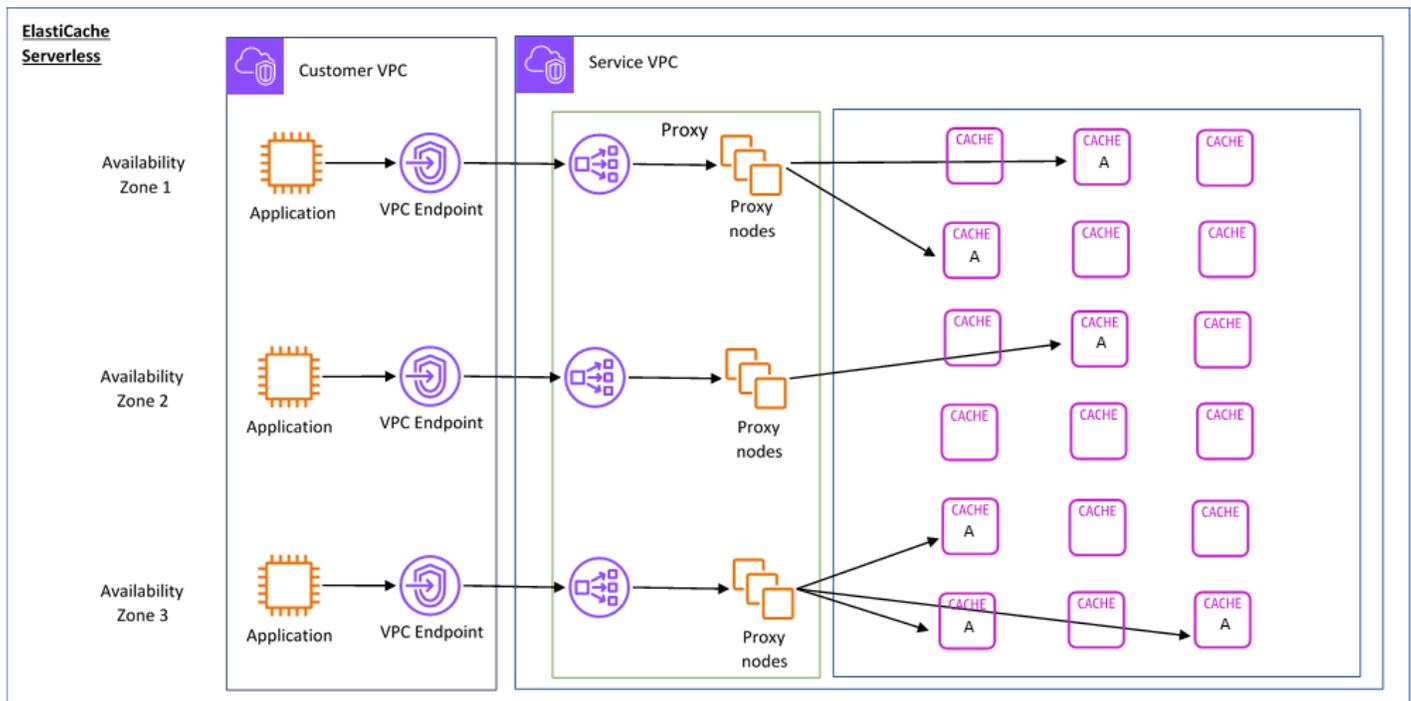
Note

- ElastiCache Serverless führt Valkey, Redis OSS oder Memcached im Clustermodus aus und ist nur mit Clients kompatibel, die dies unterstützen. TLS

Wichtigste Vorteile

- **Keine Kapazitätsplanung:** ElastiCache Serverless macht die Kapazitätsplanung überflüssig. ElastiCache Serverless überwacht kontinuierlich die Speicher-, Rechen- und Netzwerkbandbreitennutzung Ihres Caches und skaliert sowohl vertikal als auch horizontal. Dadurch kann ein Cache-Knoten an Größe zunehmen und gleichzeitig ein Aufskalierungsvorgang initiiert werden, um sicherzustellen, dass der Cache jederzeit skaliert werden kann, um Ihre Anwendungsanforderungen zu erfüllen.
- **Pay-per-use:** Bei ElastiCache Serverless zahlen Sie für die im Cache gespeicherten Daten und die Rechenleistung, die von Ihrem Workload genutzt wird. Siehe [Preisdimensionen](#).
- **Hochverfügbarkeit:** ElastiCache Serverless repliziert Ihre Daten automatisch über mehrere Availability Zones (AZ) hinweg, um eine hohe Verfügbarkeit zu gewährleisten. Es überwacht automatisch die zugrunde liegenden Cache-Knoten und ersetzt sie bei Ausfällen. Es bietet eine Verfügbarkeit von 99,99% für jeden Cache. SLA
- **Automatische Software-Upgrades:** ElastiCache Serverless aktualisiert Ihren Cache automatisch auf die neueste Minor- und Patch-Softwareversion, ohne dass die Verfügbarkeit Ihrer Anwendung beeinträchtigt wird. Sobald eine neue Hauptversion verfügbar ist, ElastiCache erhalten Sie eine Benachrichtigung.
- **Sicherheit:** Serverless verschlüsselt Daten während der Übertragung und im Ruhezustand. Sie können einen vom Service verwalteten Schlüssel oder einen eigenen kundenverwalteten Schlüssel verwenden, um Daten im Ruhezustand zu verschlüsseln.

Das folgende Diagramm zeigt, wie ElastiCache Serverless funktioniert.



Wenn Sie einen neuen serverlosen Cache erstellen, ElastiCache erstellt er einen Virtual Private Cloud (VPC) -Endpunkt in den Subnetzen Ihrer Wahl in Ihrem VPC. Ihre Anwendung kann über diese VPC Endpunkte eine Verbindung zum Cache herstellen.

Mit ElastiCache Serverless erhalten Sie einen einzigen DNS Endpunkt, mit dem Ihre Anwendung eine Verbindung herstellt. Wenn Sie eine neue Verbindung zum Endpunkt anfordern, verarbeitet ElastiCache Serverless alle Cache-Verbindungen über eine Proxyschicht. Die Proxy-Schicht trägt zu einer weniger komplexen Client-Konfiguration bei, da der Client die Cluster-Topologie nicht erneut ermitteln muss, wenn Änderungen am zugrunde liegenden Cluster vorgenommen werden. Die Proxy-Schicht besteht aus einer Reihe von Proxy-Knoten, die Verbindungen mithilfe eines Network Load Balancer verarbeiten.

Wenn Ihre Anwendung eine neue Cache-Verbindung erstellt, wird die Anfrage vom Network Load Balancer an einen Proxy-Knoten gesendet. Führt Ihre Anwendung Cache-Befehle aus, dann führt der Proxy-Knoten, der mit Ihrer Anwendung verbunden ist, die Anfragen auf einem Cache-Knoten in Ihrem Cache aus. Die Proxy-Schicht abstrahiert die Cache-Cluster-Topologie und die Knoten von Ihrem Client. Dies ermöglicht ElastiCache einen intelligenten Lastenausgleich, die Skalierung und das Hinzufügen neuer Cache-Knoten, den Austausch von Cache-Knoten bei deren Ausfall und die Aktualisierung der Software auf den Cache-Knoten, ohne dass die Verfügbarkeit Ihrer Anwendung beeinträchtigt oder Verbindungen zurückgesetzt werden müssen.

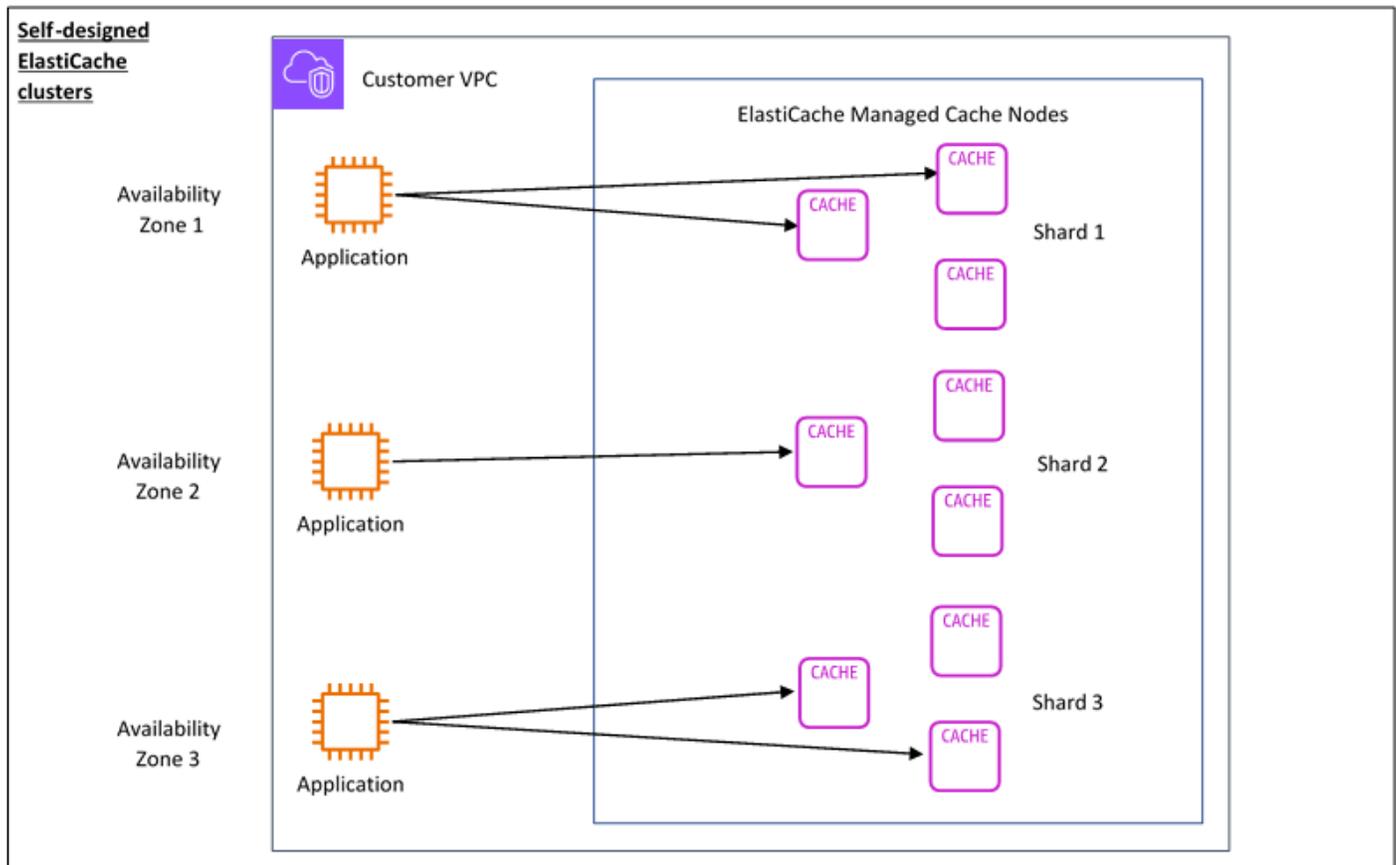
Selbst entworfene Cluster ElastiCache

Sie können Ihre eigenen ElastiCache Cluster entwerfen, indem Sie eine Cache-Knotenfamilie, Größe und Anzahl der Knoten für Ihren Cluster auswählen. Durch das Entwerfen eines eigenen Clusters haben Sie eine genauere Kontrolle und können die Anzahl der Shards in Ihrem Cache und die Anzahl der Knoten (Primärknoten und Replikatknoten) in jedem Shard auswählen. Sie können wählen, ob Sie Valkey oder Redis OSS im Clustermodus betreiben möchten, indem Sie einen Cluster mit mehreren Shards erstellen, oder im Nicht-Cluster-Modus mit einem einzigen Shard.

Wichtigste Vorteile

- Entwerfen Sie Ihren eigenen Cluster: Mit können Sie Ihren eigenen Cluster entwerfen und wählen ElastiCache, wo Sie Ihre Cache-Knoten platzieren möchten. Wenn Sie beispielsweise über eine Anwendung verfügen, bei der es um einen Kompromiss zwischen hoher Verfügbarkeit und niedriger Latenz geht, können Sie sich dafür entscheiden, Ihre Cache-Knoten in einer einzigen AZ bereitzustellen. Alternativ können Sie Ihren Cluster mit Knoten entwerfen, die sich über mehrere Knoten verteilenAZs, um eine hohe Verfügbarkeit zu erreichen.
- Genauere Kontrolle: Wenn Sie einen eigenen Cluster entwerfen, haben Sie mehr Kontrolle über die Optimierung der Einstellungen in Ihrem Cache. Beispielsweise können Sie [Valkey- und Redis-Parameter OSS](#) oder verwenden, [Memcached-spezifische Parameter](#) um die Cache-Engine zu konfigurieren.
- Vertikales und horizontales Skalieren: Sie können Ihren Cluster wahlweise manuell skalieren, indem Sie die Größe des Cache-Knotens bei Bedarf erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie neue Shards oder weitere Replikate zu Ihren Shards hinzufügen. Sie können die Auto-Scaling-Funktion auch verwenden, um die Skalierung auf der Grundlage eines Zeitplans oder eine Skalierung auf der Grundlage von Metriken wie CPU der Speichernutzung im Cache zu konfigurieren.

Das folgende Diagramm zeigt, wie ElastiCache selbst entworfene Cluster funktionieren.



Preisdimensionen

Sie können die Bereitstellung ElastiCache in zwei Bereitstellungsoptionen durchführen. Bei der Bereitstellung von ElastiCache Serverless zahlen Sie für die Nutzung der in GB-Stunden gespeicherten Daten und für die Rechenleistung in ElastiCache Verarbeitungseinheiten (ECPUs). Wenn Sie sich dafür entscheiden, Ihre eigenen ElastiCache Cluster zu entwerfen, zahlen Sie pro Stunde für die Nutzung des Cache-Knotens. Weitere Informationen zu Preisen finden Sie [hier](#).

Datenspeicher

Sie zahlen für in ElastiCache Serverless gespeicherte Daten, die in Gigabyte-Stunden (GB-Stunden) abgerechnet werden. ElastiCache Serverless überwacht kontinuierlich die in Ihrem Cache gespeicherten Daten, wobei mehrere Samplings pro Minute durchgeführt werden, und berechnet einen Stundendurchschnitt, um die Datenspeichernutzung des Caches in GB-Stunden zu ermitteln. Jeder ElastiCache serverlose Cache wird für mindestens 1 GB an gespeicherten Daten gemessen.

ElastiCache Verarbeitungseinheiten (ECPUs)

Sie zahlen für die Anfragen, die Ihre Anwendung auf ElastiCache Serverless ausführt, in ElastiCache Processing Units (ECPUs), einer Einheit, die sowohl die CPU V-Zeit als auch die übertragenen Daten umfasst.

- Einfache Lese- und Schreibvorgänge erfordern 1 ECPU pro Kilobyte (KB) übertragener Daten. Beispielsweise verbraucht ein GET Befehl, der bis zu 1 KB an Daten überträgt, 1 ECPU. Eine SET Anfrage, die 3,2 KB an Daten überträgt, verbraucht 3,2 ECPUs KB.
- Bei Valkey und Redis werden Befehle OSS, die mehr CPU V-Zeit verbrauchen und mehr Daten übertragen, auf der ECPUs Grundlage der höheren der beiden Dimensionen verbraucht. Wenn Ihre Anwendung beispielsweise den HMGET Befehl verwendet, dreimal so viel CPU Zeit wie ein einfacher SET/GET-Befehl verbraucht und 3,2 KB an Daten überträgt, verbraucht sie 3,2 ECPUs KB. Wenn sie jedoch nur 2 KB an Daten überträgt, verbraucht sie 3 ECPUs KB.
- Bei Valkey und Redis verbrauchen Befehle OSS, die zusätzliche CPU V-Zeit benötigen, proportional mehr ECPUs. Wenn Ihre Anwendung beispielsweise den OSS [HMGET Befehl](#) Valkey oder Redis verwendet und die 3-fache CPU V-Zeit wie ein einfacher SET/GET-Befehl verbraucht, dann verbraucht sie 3 ECPUs.
- Mit Memcached verbrauchen Befehle, die auf mehrere Elemente angewendet werden, proportional mehr ECPUs. Wenn Ihre Anwendung beispielsweise ein Multiget für 3 Elemente ausführt, verbraucht sie 3 Objekte ECPUs.
- Bei Memcached werden Befehle, die auf mehr Elemente angewendet und mehr Daten übertragen, auf der ECPUs Grundlage der höheren der beiden Dimensionen verbraucht. Wenn Ihre Anwendung beispielsweise den GET Befehl verwendet, 3 Elemente abrufen und 3,2 KB an Daten überträgt, verbraucht sie 3,2 ECPUs KB. Wenn sie jedoch nur 2 KB an Daten überträgt, verbraucht sie 3 ECPUs KB.

ElastiCache Serverless gibt eine neue Metrik mit dem Namen `awsElastiCacheProcessingUnits`, die Ihnen hilft, zu verstehen, wie viel Ihr ECPUs Workload verbraucht.

Knotenstunden

Sie können Ihren eigenen Cache-Cluster entwerfen, indem Sie die EC2 Knotenfamilie, Größe, Anzahl der Knoten und die Platzierung in den Availability Zones auswählen. Wenn Sie Ihren Cluster selbst entwerfen, zahlen Sie pro Stunde für jeden Cache-Knoten.

ElastiCache Backups

Ein Backup ist eine point-in-time Kopie eines serverlosen Caches oder eines von Valkey oder Redis OSS selbst entworfenen Clusters. ElastiCache ermöglicht es Ihnen, jederzeit eine Sicherungskopie Ihrer Daten zu erstellen oder automatische Backups einzurichten. Backups können zur Wiederherstellung eines bestehenden Caches oder zum Seeding eines neuen Caches verwendet werden. Backups umfassen alle Daten in einem Cache sowie zusätzlich einige Metadaten. Weitere Informationen finden Sie unter [Snapshot und Wiederherstellung](#).

Auswählen aus den Bereitstellungsoptionen

Amazon ElastiCache bietet zwei Bereitstellungsoptionen:

- Serverless-Caching
- Selbst entworfene Cluster

Eine Liste der unterstützten Befehle für beide finden Sie unter [Unterstützte und eingeschränkte Valkey-, Redis OSS - und Memcached-Befehle](#).

Serverless-Caching

Amazon ElastiCache Serverless vereinfacht die Cache-Erstellung und lässt sich sofort skalieren, um die anspruchsvollsten Anwendungen der Kunden zu unterstützen. Mit ElastiCache Serverless können Sie in weniger als einer Minute einen hochverfügbaren und skalierbaren Cache erstellen, sodass Sie keine Cache-Cluster-Kapazität bereitstellen, planen und verwalten müssen. ElastiCache Serverless speichert Daten automatisch redundant in drei Availability Zones und bietet ein Service Level Agreement (SLA) mit einer Verfügbarkeit von 99,99%. SLA Backups von selbst entworfenen Valkey- oder OSS Redis-Clustern können in einer serverlosen Konfiguration wiederhergestellt werden.

Selbst entworfene Cluster

Wenn Sie eine genaue Kontrolle über Ihren Valkey-, Redis- oder Memcached-Cluster benötigen, können Sie OSS wählen, ob Sie Ihren eigenen Cluster damit entwerfen möchten. ElastiCache ermöglicht Ihnen den Betrieb eines knotenbasierten Clusters, indem Sie den Knotentyp, die Anzahl der Knoten und die Platzierung der Knoten in den Availability Zones für Ihren Cluster auswählen. AWS Da es sich um einen vollständig verwalteten Service ElastiCache handelt, hilft er bei der Verwaltung der Hardwarebereitstellung, der Überwachung, des Austauschs von Knoten und des Software-Patches für Ihren Cluster. Selbst entworfene Cluster können so konzipiert werden, dass sie

eine Verfügbarkeit von bis zu 99,99% bieten. SLA Backups aus serverlosen Valkey- oder OSS Redis-Caches können in einem selbst entworfenen Cluster wiederhergestellt werden.

Auswählen aus den Bereitstellungsoptionen

Wählen Sie Serverless-Caching in folgenden Fällen:

- Sie erstellen einen Cache für Workloads, die entweder neu oder schwer vorhersehbar sind.
- Der Datenverkehr Ihrer Anwendung ist nicht berechenbar.
- Sie möchten den einfachsten Einstieg in die Arbeit mit einem Cache.

Entscheiden Sie sich dafür, Ihren eigenen ElastiCache Cluster zu entwerfen, wenn:

- Sie verwenden bereits ElastiCache Serverless und möchten eine genauere Kontrolle über den Knotentyp, auf dem Valkey, Redis oder Memcached ausgeführt werden OSS, die Anzahl der Knoten und die Platzierung dieser Knoten.
- Sie erwarten, dass Ihr Anwendungsdatenverkehr relativ vorhersehbar ist, und Sie möchten eine genaue Kontrolle über Leistung, Verfügbarkeit und Kosten.
- Sie können Ihre Kapazitätsanforderungen vorhersagen, um Kosten zu kontrollieren.

Vergleich von serverlosem Caching und selbst entworfenen Clustern

Funktion	Serverless-Caching	Selbst entworfene Cluster
Cache-Einrichtung	Erstellen Sie in weniger als einer Minute einen Cache mit nur einem Namen	Bietet eine detaillierte Kontrolle über das Cache-Cluster-Design. Der Benutzer kann den Knotentyp, die Anzahl der Knoten und die Platzierung in den Availability Zones wählen AWS
Unterstützte Version ElastiCache	Valkey 7.2 und höher, Redis OSS Version 7.1 und höher, Memcached 1.6.21 und höher	Valkey 7.2 und höher, Redis OSS Version 4.0 und höher, Memcached 1.4 und höher

Funktion	Serverless-Caching	Selbst entworfene Cluster
Cluster-Modus (Valkey und Redis) OSS	Betreibt Motoren nur <code>incluster mode enabled</code> . Clients müssen die Verbindung <code>cluster mode enabled</code> zu ElastiCache Serverless unterstützen.	Kann so konfiguriert werden, dass der Clustermodus aktiviert oder der Clustermodus deaktiviert ist.
Skalierung	Skaliert Engines automatisch sowohl vertikal als auch horizontal ohne Kapazitätsmanagement.	<p>Ermöglicht die Kontrolle über die Skalierung und erfordert gleichzeitig eine Überwachung, um sicherzustellen, dass die aktuelle Kapazität dem Bedarf angemessen entspricht.</p> <p>Bei Valkey und Redis können Sie wählen OSS, ob Sie vertikal skalieren möchten, indem Sie die Größe des Cache-Knotens bei Bedarf erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie neue Shards oder weitere Replikate zu Ihren Shards hinzufügen. Diese Funktion ist für Memcached nicht verfügbar.</p> <p>Mit der Auto-Scaling-Funktion können Sie die Skalierung auch auf der Grundlage eines Zeitplans oder anhand von Messwerten wie der CPU Speicherauslastung im Cache konfigurieren.</p>

Funktion	Serverless-Caching	Selbst entworfene Cluster
Client-Verbindung	Clients stellen eine Verbindung zu einem einzelnen Endpunkt her. Dadurch kann die zugrunde liegende Cache-Knotentopologie (Skalierung, Austausch und Upgrades) geändert werden, ohne dass die Verbindung zum Client unterbrochen werden muss.	Clients stellen eine Verbindung zu jedem einzelnen Cache-Knoten her. Wenn ein Knoten ersetzt wird, erkennt der Client die Clustertopologie erneut und stellt die Verbindungen wieder her.
Konfigurierbarkeit	Keine detaillierte Konfiguration verfügbar. Kunden können grundlegende Einstellungen konfigurieren, darunter Subnetze, die auf den Cache zugreifen können, ob automatische Backups aktiviert oder deaktiviert werden, und maximale Cache-Nutzungsbeschränkungen.	Selbst entworfene Cluster bieten detaillierte Konfigurationsoptionen. Kunden können Parametergruppen für eine detaillierte Steuerung verwenden. Eine Tabelle dieser Parameterwerte nach Knotentyp finden Sie unter Motorspezifische Parameter .
Multi-AZ	Daten werden asynchron über mehrere Availability Zones hinweg repliziert, um eine höhere Verfügbarkeit und eine verbesserte Leselatenz zu erreichen.	Bietet eine Option zum Entwerfen des Clusters in einer einzelnen Availability Zone oder über mehrere Availability Zones hinweg (AZs). Bietet bei Verwendung von Valkey oder Redis Multi-AZ-Cluster mit DatenOSS, die asynchron über mehrere Availability Zones repliziert werden, um eine höhere Verfügbarkeit und eine verbesserte Leselatenz zu erzielen.

Funktion	Serverless-Caching	Selbst entworfene Cluster
Verschlüsselung im Ruhezustand	Immer aktiviert. Kunden können einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel in verwenden AWS KMS.	Option zum Aktivieren oder Deaktivieren der Verschlüsselung im Ruhezustand. Wenn diese Option aktiviert ist, können Kunden einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel in verwenden AWS KMS.
Verschlüsselung bei der Übertragung (TLS)	Immer aktiviert. Clients müssen TLS Konnektivität unterstützen.	Option zum Aktivieren oder Deaktivieren.
Sicherungen	<p>Unterstützt automatische und manuelle Backups von Caches ohne Leistungs einbußen.</p> <p>Valkey- und OSS Redis-Backups sind untereinander kompatibel und können in einem ElastiCache serverlosen Cache oder einem selbst entworfenen Cluster wiederhergestellt werden.</p>	<p>Unterstützt automatische und manuelle Backups für Valkey und Redis. OSS Bei Clustern kann es je nach verfügbarem reservierten Speicher zu Leistungseinbußen kommen. Weitere Informationen finden Sie unter Verwaltung des reservierten Speichers für Valkey und Redis OSS.</p> <p>Valkey- und OSS Redis-Backups sind untereinander kompatibel und können in einem ElastiCache serverlosen Cache oder einem selbst entworfenen Cluster wiederhergestellt werden.</p>

Funktion	Serverless-Caching	Selbst entworfene Cluster
Überwachen	<p>Support Metriken auf Cache-Ebene, einschließlich Cache-Trefferquote, Cache-Fehlrate, Datengröße und ECPUs verbrauchter Daten.</p> <p>ElastiCache Serverless sendet Ereignisse und verwendet dabei den EventBridge Zeitpunkt, an dem signifikante Ereignisse in Ihrem Cache auftreten. Sie können wählen, ob Sie ElastiCache Ereignisse mithilfe von Amazon EventBridge überwachen, aufnehmen, transformieren und darauf reagieren möchten. Weitere Informationen finden Sie unter Serverless-Cache-Ereignisse.</p>	<p>ElastiCache Selbst entworfene Cluster geben auf jeder Knotenebene Metriken aus, darunter sowohl Metriken auf Host-Ebene als auch Cache-Metriken.</p> <p>Selbst entworfene Cluster senden SNS Benachrichtigungen für wichtige Ereignisse aus. Siehe Metriken für Memcached und Metriken für Valkey und Redis OSS.</p>
Verfügbarkeit	<p>Service Level Agreement () mit einer Verfügbarkeit von 99,99% SLA</p>	<p>Selbst entworfene Cluster können so konzipiert werden, dass sie je nach Konfiguration ein Service Level Agreement (SLA) mit einer Verfügbarkeit von bis zu 99,99% erreichen.</p>

Funktion	Serverless-Caching	Selbst entworfene Cluster
Software-Upgrades und Patches	Führt ein automatisches Upgrade der Cache-Software auf die neueste Minor- und Patch-Version durch, ohne dass sich dies auf die Anwendung auswirkt. Kunden erhalten eine Benachrichtigung über Hauptversions-Upgrades, und Kunden können jederzeit auf die neueste Hauptversion aktualisieren.	Selbst entworfene Cluster bieten kundenseitig aktivierten Self-Service für kleinere und patchende Versions-Upgrades sowie für Hauptversions-Upgrades. Verwaltete Updates werden automatisch während der vom Kunden definierten Wartungsfenster installiert. Kunden können auch wählen, ob sie bei Bedarf ein Upgrade auf eine Nebenversion oder eine Patch-Version durchführen möchten.
Globaler Datenspeicher	Nicht unterstützt	Unterstützt Global Data Store, der eine regionsübergreifende Replikation mit Schreibvorgängen in einer Region und Lesevorgängen in mehreren Regionen ermöglicht

Funktion	Serverless-Caching	Selbst entworfene Cluster
Daten-Tiering	Nicht unterstützt	Bei Clustern, die mit Knoten der R6GD-Familie entworfen wurden, werden die Daten zwischen Arbeitsspeicher und lokalem Speicher SSD (Solid-State-Laufwerke) aufgeteilt. Data Tiering bietet eine preisgünstige Option für Valkey- und OSS Redis-Workloads, da neben der Speicherung von Daten im Arbeitsspeicher auch kostengünstigere Solid-State-Laufwerke (SSDs) in jedem Clusterknoten verwendet werden.
Preismodell	Pay-per-use, basierend auf Daten, die in GB-Stunden gespeichert sind, und Anfragen in Verarbeitungseinheiten (V). ElastiCache ECPU Weitere Informationen zu Preisen finden Sie hier .	Pay-per-hour, basierend auf der Nutzung des Cache-Knotens. Weitere Informationen zu Preisen finden Sie hier .

Verwandte Themen:

- [Entwerfen und Verwalten Ihres eigenen ElastiCache Clusters](#)

ElastiCache Amazon-Ressourcen für Erstnutzer

Wir empfehlen Erstbenutzern, zunächst die folgenden Abschnitte zu lesen und sie bei Bedarf zu lesen.

- Leistungsmerkmale und Preise — Die [Produktdetailseite](#) bietet einen allgemeinen Produktüberblick über ElastiCache die wichtigsten Services und die Preisgestaltung.
- ElastiCache Videos — [ElastiCache Videos](#) In diesem Bereich finden Sie Videos, die Ihnen Amazon vorstellen ElastiCache. Die Videos behandeln allgemeine Anwendungsfälle ElastiCache und zeigen, wie Sie ElastiCache damit die Latenz reduzieren und den Durchsatz Ihrer Anwendungen verbessern können.
- Erste Schritte – Der [Erste Schritte mit Amazon ElastiCache](#)-Abschnitt enthält Informationen zur Erstellung eines Cache-Clusters. Das Beispiel umfasst auch die Autorisierung des Zugriffs auf den Cache-Cluster, die Verbindung mit einem Cache-Knoten und das Löschen eines Cache-Clusters.
- Skalierbare Leistung — Das ElastiCache Whitepaper [Performance at Scale with Amazon befasst sich mit](#) Caching-Strategien, die dazu beitragen, dass Ihre Anwendung im großen Maßstab eine gute Leistung erbringt.

Wenn Sie die vorhergehenden Abschnitte abgeschlossen haben, lesen die folgenden:

- [Auswahl der Knotengröße](#)

Knoten müssen so groß dimensioniert sein, dass sie alle Daten speichern können, die in den Cache aufgenommen werden sollen. Dabei möchten Sie auch nicht für mehr Cache bezahlen, als Sie benötigen. In diesem Thema finden Sie Informationen zur Auswahl der optimalen Knotengröße.

- [ElastiCache bewährte Methoden und Caching-Strategien](#)

Identifizieren und Beheben von Problemen, die sich auf die Effizienz Ihres Clusters auswirken können.

Wenn Sie das AWS Command Line Interface (AWS CLI) verwenden möchten, können Sie die folgenden Dokumente verwenden, um Ihnen den Einstieg zu erleichtern:

- [AWS Command Line Interface Dokumentation](#)

Dieser Abschnitt enthält Informationen zum Herunterladen von AWS CLI, zum AWS CLI Betrieb auf Ihrem System und zum Eingeben Ihrer AWS Anmeldeinformationen.

- [AWS CLI Dokumentation für ElastiCache](#)

Dieses separate Dokument behandelt alle ElastiCache Befehle AWS CLI für, einschließlich Syntax und Beispielen.

Sie können Anwendungsprogramme schreiben, um sie ElastiCache API mit einer Vielzahl gängiger Programmiersprachen zu verwenden. Nachfolgend finden Sie einige Ressourcen:

- [Tools für Amazon Web Services](#)

Amazon Web Services bietet eine Reihe von Software Development Kits (SDKs) mit Unterstützung für ElastiCache. Sie können für die ElastiCache Verwendung von Java programmieren, .NET, PHP, Ruby und andere Sprachen. Diese SDKs können Ihre Anwendungsentwicklung erheblich vereinfachen, indem sie Ihre Anfragen formatieren ElastiCache, Antworten analysieren und Wiederholungslogik und Fehlerbehandlung bereitstellen.

- [Mit dem ElastiCache API](#)

Wenn Sie die nicht verwenden möchten AWS SDKs, können Sie ElastiCache direkt mit der Abfrage interagieren. API In diesem Abschnitt finden Sie auch Tipps zur Fehlerbehebung sowie Informationen zur Erstellung und Authentifizierung von Anfragen und der Behandlung von Antworten.

- [ElastiCache API Amazon-Referenz](#)

Dieses separate Dokument behandelt alle ElastiCache API Operationen, einschließlich Syntax und Beispielen.

AWS Regionen und Verfügbarkeitszonen

Amazon Cloud Computing-Ressourcen sind in hochverfügbaren Rechenzentren in verschiedenen Regionen der Welt untergebracht (zum Beispiel in Nordamerika, Europa oder Asien). Jeder Rechenzentrumsstandort wird als AWS Region bezeichnet.

Jede AWS Region enthält mehrere unterschiedliche Standorte, die als Availability Zones oder bezeichnet AZs werden. Jede Availability Zone ist so aufgebaut, dass sie von Fehlern in anderen Availability Zones nicht betroffen ist. Jede ist so konzipiert, dass sie kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben AWS Region bietet. Indem Instances in separaten Availability Zones gestartet werden, können Sie Ihre Anwendungen vor den Fehlern eines einzelnen Standorts schützen. Weitere Informationen finden Sie unter [Auswählen von Regionen und Availability Zones](#).

Sie können Ihren Cluster in mehreren Availability Zones erstellen, eine Option, die als Multi-AZ-Bereitstellung bezeichnet wird. Wenn Sie diese Option wählen, stellt Amazon automatisch eine sekundäre Standby-Knoten-Instance in einer anderen Availability Zone bereit und verwaltet sie. Ihre

Primärknoten-Instance wird asynchron über Availability Zones auf die sekundäre Instance repliziert. Dieser Ansatz sorgt für Datenredundanz und Failover-Unterstützung, vermeidet das Einfrieren von I/O-Vorgängen und minimiert Latenzspitzen im Verlauf von Systemsicherungen. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache \(RedisOSS\)](#) mit Multi-AZ.

Häufige ElastiCache Anwendungsfälle und wie ElastiCache Sie helfen können

Ob es sich um die neuesten Nachrichten, eine Top-10-Rangliste, einen Produktkatalog oder den Verkauf von Eintrittskarten für eine Veranstaltung handelt - Geschwindigkeit ist das A und O. Der Erfolg Ihrer Website und Ihres Unternehmens hängt stark von der Geschwindigkeit ab, mit der Sie Inhalte bereitstellen.

In dem Artikel "[Für ungeduldige Internetnutzer ist ein Augenzwinkern einfach zu lang, um zu warten](#)" stellte die New York Times fest, dass die Nutzer einen Unterschied von 250 Millisekunden (1/4 Sekunde) zwischen konkurrierenden Websites registrieren können. Die Nutzer neigen dazu, die langsamere Seite zugunsten der schnelleren Seite zu verlassen. Tests bei Amazon, die in dem Artikel [Wie die Ladezeit von Webseiten mit dem Verlust von Besuchern zusammenhängt](#) zitiert werden, ergaben, dass für jede Erhöhung der Ladezeit um 100 ms (1/10 Sekunde) die Verkaufszahlen um 1 Prozent sinken.

Wenn jemand Daten wünscht, können Sie diese Daten viel schneller bereitstellen, wenn sie im Cache gespeichert sind. Das gilt sowohl für eine Webseite als auch für einen Bericht, der als Grundlage für Geschäftsentscheidungen dient. Kann es sich Ihr Unternehmen leisten, Ihre Webseiten nicht zwischenspeichern, um sie so mit der kürzestmöglichen Latenz bereitzustellen?

Es mag intuitiv einleuchtend erscheinen, dass Sie die am häufigsten nachgefragten Artikel in den Cache stellen wollen. Aber warum sollten Sie nicht auch weniger häufig angefragte Elemente zwischenspeichern? Selbst die optimierteste Datenbankabfrage oder der am besten optimierte API Remoteaufruf ist deutlich langsamer als das Abrufen eines Flatkeys aus einem In-Memory-Cache. Eine merkliche Verlangsamung führt dazu, dass Kunden abwandern.

Die folgenden Beispiele veranschaulichen einige Möglichkeiten, wie die Verwendung die Gesamtleistung Ihrer Anwendung verbessern ElastiCache kann.

Themen

- [In-Memory-Datastore](#)
- [Bestenlisten für Gaming-Spiele](#)
- [Nachrichtenübermittlung \(Pub/Sub\)](#)
- [Empfehlungsdaten \(Hashes\)](#)
- [ElastiCache Kundenreferenzen](#)

In-Memory-Datastore

Der primäre Zweck eines Hauptspeicher-basierten Key-Value Store besteht darin, superschnellen (mit einer Latenz von unter einer Millisekunde) und kostengünstigen Zugriff auf Kopien von Daten bereitzustellen. Die meisten Datenspeicher haben Bereiche mit Daten, auf die häufig zugegriffen wird, die aber selten aktualisiert werden. Außerdem wird das Abfragen einer Datenbank immer langsamer und kostspieliger sein als das Auffinden eines Schlüssels in einem Schlüsselwertpaar-Cache. Manche Datenbankabfragen sind besonders aufwendig in der Durchführung. Ein Beispiel dafür sind Abfragen, die Joins über mehrere Tabellen oder Abfragen mit intensiven Berechnungen beinhalten. Durch die Zwischenspeicherung solcher Abfrageergebnisse zahlen Sie den Preis für die Abfrage nur einmal. Dann können Sie die Daten schnell und mehrfach abrufen, ohne die Abfrage erneut ausführen zu müssen.

Was sollte ich zwischenspeichern?

Bei der Entscheidung, welche Daten zwischengespeichert werden, sind folgende Faktoren zu berücksichtigen:

Geschwindigkeit und Kosten – Es ist immer langsamer und teurer, Daten aus einer Datenbank abzurufen als aus einem Cache. Manche Datenbankabfragen sind grundsätzlich langsamer und kostspieliger als andere. Beispielsweise sind Abfragen, bei denen mehrere Tabellen miteinander verknüpft werden, viel langsamer und teurer als einfache Abfragen, die nur eine Tabelle betreffen. Wenn die interessanten Daten eine langsame und teure Abfrage erfordern, sind sie ein Kandidat für das Caching. Wenn der Abruf der Daten eine relativ schnelle und einfache Abfrage erfordert, kann er je nach anderen Faktoren immer noch ein Kandidat für die Zwischenspeicherung sein.

Daten und Zugriffsmuster – Um zu bestimmen, was zwischengespeichert werden soll, müssen auch die Daten selbst und ihre Zugriffsmuster verstanden werden. So ist es beispielsweise nicht sinnvoll, Daten, die sich schnell ändern oder auf die nur selten zugegriffen wird, im Cache zu speichern. Damit die Zwischenspeicherung einen echten Nutzen bringt, sollten die Daten relativ statisch sein und häufig abgerufen werden. Ein Beispiel wäre ein persönliches Profil auf einer Social-Media-Website. Andererseits sollten Sie keine Daten zwischenspeichern, wenn das Zwischenspeichern keine Geschwindigkeits- oder Preisvorteile bringt. So ist es beispielsweise nicht sinnvoll, Webseiten, die Suchergebnisse liefern, in den Cache zu stellen, da die Abfragen und Ergebnisse in der Regel einzigartig sind.

Unvergänglichkeit – Per Definition sind zwischengespeicherte Daten veraltete Daten. Selbst wenn es unter bestimmten Umständen nicht veraltet ist, sollte es immer als veraltet betrachtet und behandelt

werden. Um festzustellen, ob Ihre Daten für die Zwischenspeicherung geeignet sind, müssen Sie die Toleranz Ihrer Anwendung gegenüber veralteten Daten ermitteln.

Es kann sein, dass Ihre Anwendung veraltete Daten in dem einen Kontext tolerieren kann, nicht jedoch in einem anderen. Nehmen wir zum Beispiel an, dass Ihre Website einen öffentlich gehandelten Aktienkurs anbietet. Ihre Kunden könnten eine gewisse Unbeständigkeit akzeptieren, wenn sie darauf hingewiesen werden, dass sich die Preise um n Minuten verzögern können. Aber wenn Sie diesen Aktienkurs einem Broker für einen Kauf oder Verkauf zur Verfügung stellen, benötigen Sie Echtzeitdaten.

Erwägen Sie die Zwischenspeicherung Ihrer Daten, wenn Folgendes zutrifft:

- Der Abruf Ihrer Daten ist im Vergleich zum Abruf aus dem Cache langsam oder teuer.
- Benutzer greifen häufig auf Ihre Daten zu.
- Ihre Daten bleiben relativ konstant, oder wenn sie sich schnell ändern, ist die Unbeständigkeit kein großes Problem.

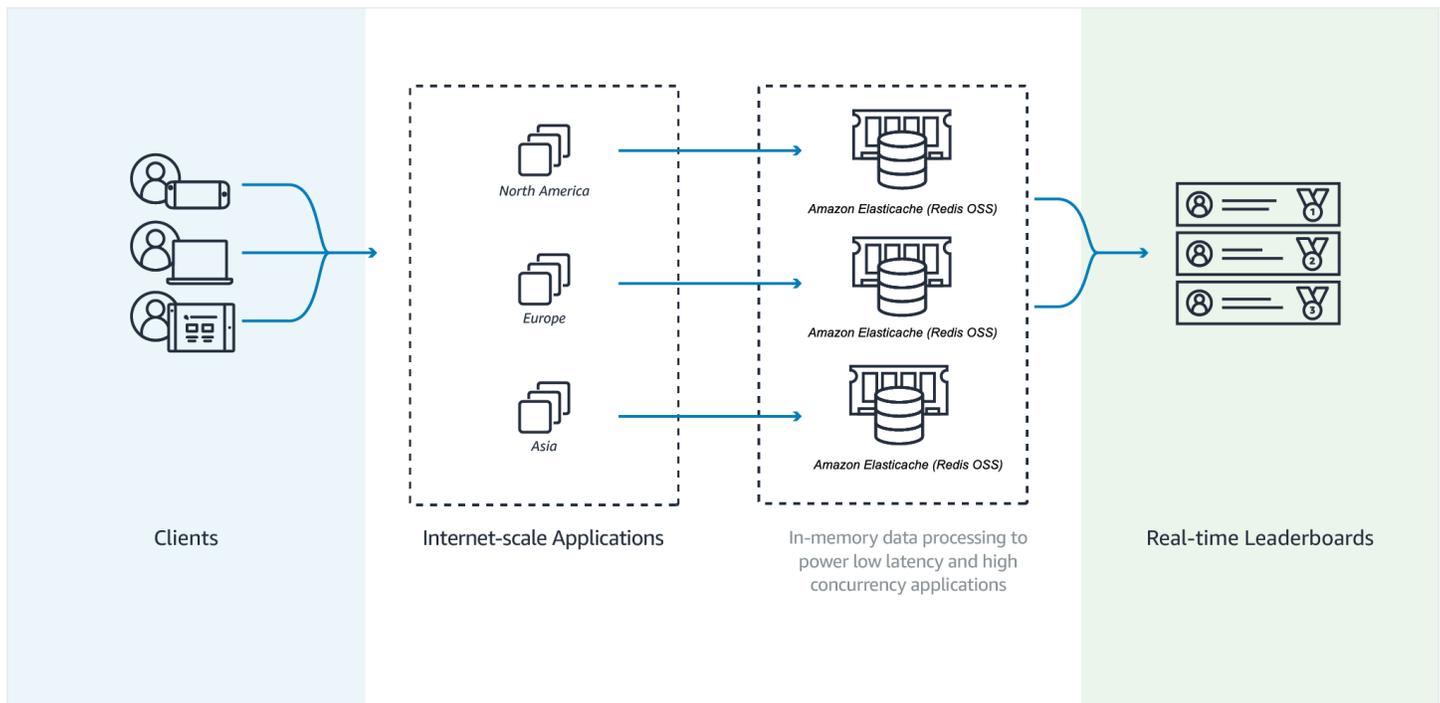
Weitere Informationen finden Sie unter [Caching-Strategien für Memcached](#)

Bestenlisten für Gaming-Spiele

Mit den OSS sortierten Sets von Valkey oder Redis können Sie die Rechenkomplexität von Bestenlisten von Ihrer Anwendung auf Ihren Cluster verlagern.

Bestenlisten, wie z. B. die 10 besten Ergebnisse eines Spiels, sind rechenintensiv. Dies gilt vor allem bei einer großen Anzahl gleichzeitiger Spieler und sich ständig ändernden Spielständen. Die OSS sortierten Sets von Valkey und Redis garantieren sowohl Einzigartigkeit als auch Reihenfolge der Elemente. Bei sortierten Mengen wird jedes Mal, wenn ein neues Element zur sortierten Menge hinzugefügt wird, es in Echtzeit neu eingestuft. Es wird dann in der numerisch richtigen Reihenfolge in die Menge eingefügt.

In der folgenden Abbildung können Sie sehen, wie eine ElastiCache Gaming-Bestenliste funktioniert.



Example Valkey- oder Redis-Bestenliste OSS

In diesem Beispiel werden vier Spieler und ihre Ergebnisse mithilfe von ZADD in eine sortierte Liste eingegeben. Der Befehl ZREVRANGEBYSCORE listet die Spieler in absteigender Reihenfolge entsprechend ihrem Ergebnis auf. Als Nächstes wird ZADD verwendet, um das Ergebnis von June zu aktualisieren, indem der bestehende Eintrag überschrieben wird. Zum Schluss werden die Spieler von ZREVRANGEBYSCORE nach ihrer Punktzahl aufgelistet, von hoch bis niedrig. Die Liste zeigt, dass June in der Rangliste aufgestiegen ist.

```
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

Mit dem folgenden Befehl erfährt June, auf welchem Platz sie unter allen Spielern steht. Da das Ranking auf Null basiert, wird für June, der an zweiter Stelle steht, eine 1 ZREVRANK zurückgegeben.

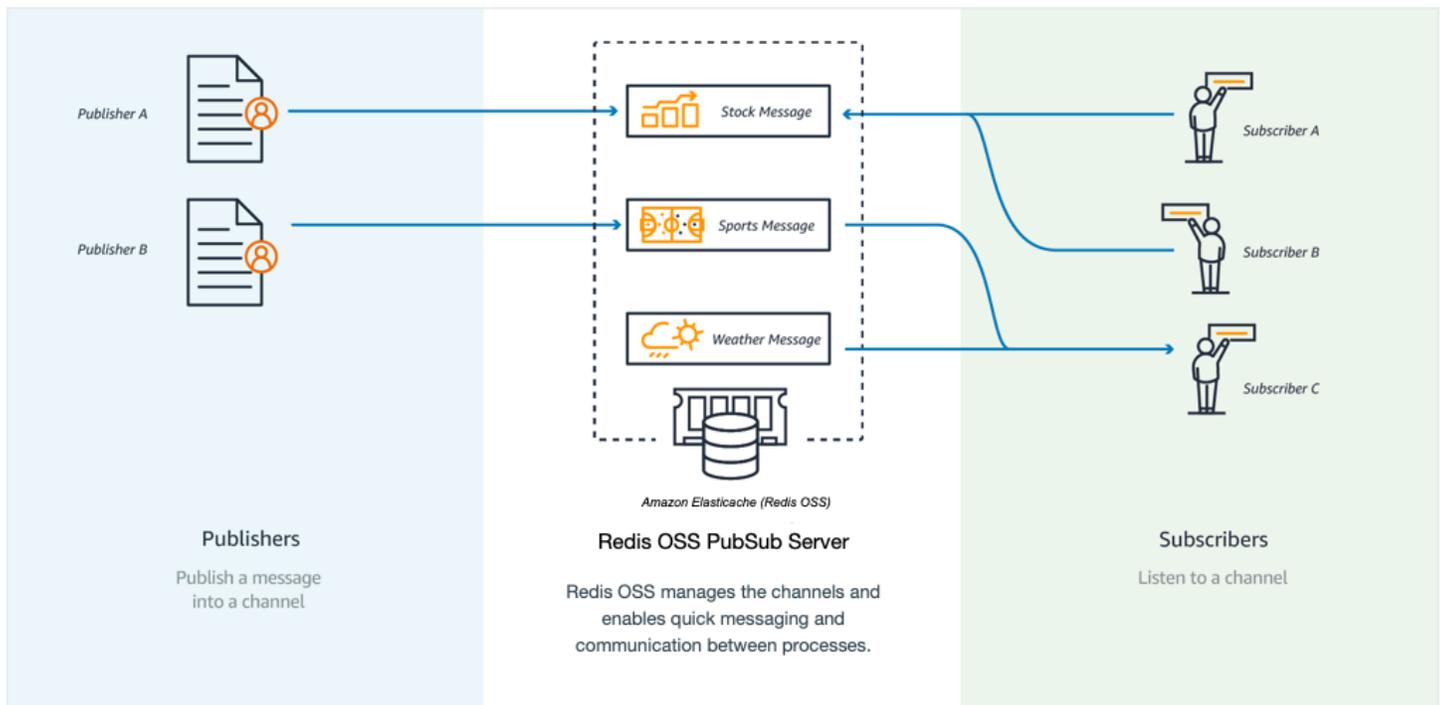
```
ZREVRANK leaderboard June  
1
```

Weitere Informationen finden Sie in der [Valkey-Dokumentation](#) zu sortierten Mengen.

Nachrichtenübermittlung (Pub/Sub)

Wenn Sie eine E-Mail-Nachricht versenden, senden Sie sie an mindestens einen angegebenen Empfänger. Im OSS Pub/Sub-Paradigma von Valkey und Redis senden Sie eine Nachricht an einen bestimmten Kanal, ohne zu wissen, wer, wenn überhaupt, sie empfängt. Die Nachricht wird nur an die Personen gesendet, die den Kanal abonniert haben. Angenommen, Sie haben etwa den news.sports.golf-Channel abonniert. Sie und alle anderen, die den Kanal news.sports.golf abonniert haben, erhalten alle auf news.sports.golf veröffentlichten Nachrichten.

Die Pub/Sub-Funktionalität hat nichts mit einem Schlüsselraum zu tun. Daher gibt es auf keinem Level Beeinträchtigungen. In der folgenden Abbildung finden Sie eine Abbildung des ElastiCache Messagings mit Valkey und Redis. OSS



Abonnieren

Um Nachrichten in einem Kanal zu erhalten, abonnieren Sie den Kanal. Sie können einen einzelnen Channel, mehrere spezielle Channels oder alle Channels, die einem Muster entsprechen, abonnieren. Um ein Abonnement zu kündigen, melden Sie sich von dem Kanal ab, den Sie abonniert haben. Wenn Sie sich mit Hilfe eines Musterabgleichs angemeldet haben, können Sie sich auch mit demselben Muster wieder abmelden, das Sie zuvor verwendet haben.

Example – Abonnement eines einzelnen Channels

Um einen einzelnen Kanal zu abonnieren, verwenden Sie den SUBSCRIBE Befehl, der den Kanal angibt, den Sie abonnieren möchten. Im folgenden Beispiel abonniert ein Client den Channel `news.sports.golf`.

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

Nach einer Weile kündigt der Client sein Abonnement für den Kanal mithilfe des UNSUBSCRIBE Befehls, der den Kanal angibt, von dem das Abonnement beendet werden soll.

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

Example – Abonnement mehrerer ausgewählter Channels

Um mehrere spezifische Kanäle zu abonnieren, listen Sie die Kanäle mit dem SUBSCRIBE Befehl auf. Im folgenden Beispiel abonniert ein Client die Channels news.sports.golf, news.sports.soccer und news.sports.skiing.

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

Um ein Abonnement für einen bestimmten Kanal zu kündigen, verwenden Sie den UNSUBSCRIBE Befehl und geben Sie den Kanal an, von dem Sie sich abmelden möchten.

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

Um Abonnements für mehrere Kanäle zu kündigen, verwenden Sie den UNSUBSCRIBE Befehl und geben Sie die Kanäle an, von denen Sie sich abmelden möchten.

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

Um alle Abonnements zu kündigen, verwenden Sie UNSUBSCRIBE und geben Sie jeden Kanal an. Oder verwenden Sie UNSUBSCRIBE ohne einen Kanal anzugeben.

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

or

```
UNSUBSCRIBE
```

Example – Abonnements mithilfe von Musterabgleich

Kunden können mit dem PSUBSCRIBE Befehl alle Kanäle abonnieren, die einem Muster entsprechen.

Im folgenden Beispiel abonniert ein Client alle Sport-Channels. Sie listen nicht alle Sportkanäle einzeln auf, wie Sie es bei SUBSCRIBE tun. Mit dem PSUBSCRIBE-Befehl verwenden Sie stattdessen den Mustervergleich.

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

Example Kündigen von Abonnements

Um Abonnements für diese Channels zu kündigen, verwenden Sie den PUNSUBSCRIBE-Befehl.

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

Important

Die Kanalzeichenfolge, die an einen [P] SUBSCRIBE -Befehl und an den Befehl [P] gesendet UNSUBSCRIBE wird, muss übereinstimmen. PSUBSCRIBE zu news.* und PUNSUBSCRIBE von news.sports.* oder UNSUBSCRIBE von news.sports.golf ist nicht möglich.

Veröffentlichen

Um eine Nachricht an alle Abonnenten eines Kanals zu senden, verwenden Sie den PUBLISH-Befehl, wobei Sie den Kanal und die Nachricht angeben. Im folgenden Beispiel wird diese Nachricht veröffentlicht: "Es ist Samstag und sonnig. Ich fahre zu den Links." auf den news.sports.golf-Kanal.

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

Ein Client kann nicht auf einem Kanal veröffentlichen, den er abonniert hat.

Weitere Informationen finden Sie unter [Pub/Sub](#) in der Valkey-Dokumentation.

Empfehlungsdaten (Hashes)

Die Verwendung von INCR oder DECR in Valkey oder Redis OSS macht das Kompilieren von Empfehlungen einfach. Immer dann, wenn Benutzer ein Produkt " liken", erhöhen Sie einen item:productID:like-Zähler. Immer dann, wenn Benutzer ein Produkt ein "disliken", erhöhen Sie einen item:productID:dislike-Zähler. Mithilfe von Hashes können Sie auch eine Liste aller Personen führen, denen ein Produkt gefallen oder nicht gefallen hat.

Example – Likes und Dislikes

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

ElastiCache Kundenreferenzen

Weitere Informationen darüber, wie Unternehmen wie AirbnbPBS, Esri und andere Amazon nutzen, ElastiCache um ihr Geschäft durch ein verbessertes Kundenerlebnis auszubauen, finden Sie unter [So nutzen andere Amazon ElastiCache](#).

Sie können sich auch die [Tutorial-Videos](#) für weitere Anwendungsfälle von ElastiCache Kunden ansehen.

Erste Schritte mit Amazon ElastiCache

Verwenden Sie das praktische Tutorial in diesem Abschnitt, um Ihnen den Einstieg zu erleichtern und mehr über die Verwendung ElastiCache zu erfahren.

Themen

- [einrichten ElastiCache](#)
- [Erstellen Sie einen serverlosen Valkey-Cache](#)
- [Erstellen Sie einen serverlosen Valkey- oder Redis-Cache OSS](#)
- [Einen serverlosen Memcached-Cache erstellen](#)
- [Tutorials: Erste Schritte mit Python und ElastiCache](#)
- [Tutorial: Konfiguration von Lambda für den Zugriff ElastiCache in einem VPC](#)

einrichten ElastiCache

Gehen Sie wie folgt vor, um den ElastiCache Webservice zu verwenden.

Themen

- [Melden Sie sich an für ein AWS-Konto](#)
- [Erstellen eines Benutzers mit Administratorzugriff](#)
- [Erteilen programmgesteuerten Zugriffs](#)
- [Richten Sie Ihre Berechtigungen ein \(nur ElastiCache für neue Benutzer\)](#)
- [Einrichten EC2](#)
- [Gewähren Sie Netzwerkzugriff von einer VPC Amazon-Sicherheitsgruppe auf Ihren Cache](#)
- [Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein](#)

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/die-Anmeldung>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff erfordern](#).

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu <https://aws.amazon.com/> gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für Ihren Root-Benutzer.

Anweisungen finden Sie im Benutzerhandbuch unter Aktivieren eines virtuellen MFA Geräts für Ihren AWS-Konto IAM Root-Benutzer ([Konsole](#)).

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) im AWS IAM Identity Center Benutzerhandbuch.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM Identity Center-Benutzer anzumelden, verwenden Sie die Anmeldung, URL die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM Identity Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal](#).

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen hierzu finden Sie unter [Berechtigungssatz erstellen](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Eine genaue Anleitung finden Sie unter [Gruppen hinzufügen](#) im AWS IAM Identity Center Benutzerhandbuch.

Erteilen programmgesteuerten Zugriffs

Benutzer benötigen programmatischen Zugriff, wenn sie mit AWS außerhalb des interagieren möchten. AWS Management Console Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift. AWS

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
<p>Mitarbeiteridentität</p> <p>(In IAM Identity Center verwaltete Benutzer)</p>	<p>Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an AWS CLI AWS SDKs, oder AWS APIs zu signieren.</p>	<p>Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.</p> <ul style="list-style-type: none"> • Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zur Verwendung AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch.
<p>IAM</p>	<p>Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder AWS APIs zu signieren.</p>	<p>Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen im IAM Benutzerhandbuch.</p>
<p>IAM</p>	<p>(Nicht empfohlen)</p> <p>Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder AWS APIs zu signieren.</p>	<p>Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.</p> <ul style="list-style-type: none"> • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM Benutzeranmeldedaten im AWS

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		<p>Command Line Interface Benutzerhandbuch.</p> <ul style="list-style-type: none"> • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch. • Weitere Informationen finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM Benutzer im IAM Benutzerhandbuch. AWS APIs

Verwandte Themen:

- [Was steht IAM](#) im IAM Benutzerhandbuch.
- AWS Allgemeine Referenz [zu Sicherheitsnachweisen](#).

Richten Sie Ihre Berechtigungen ein (nur ElastiCache für neue Benutzer)

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Benutzer, IAM die über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Folgen Sie den Anweisungen [unter Erstellen einer Rolle für einen externen Identitätsanbieter \(Federation\)](#) im IAM Benutzerhandbuch.

- IAM Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen [unter Eine Rolle für einen IAM Benutzer erstellen](#) im IAMBenutzerhandbuch.
- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Folgen Sie den Anweisungen [unter Hinzufügen von Berechtigungen für einen Benutzer \(Konsole\)](#) im IAMBenutzerhandbuch.

Amazon ElastiCache erstellt und verwendet servicebezogene Rollen, um in Ihrem Namen Ressourcen bereitzustellen und auf andere AWS Ressourcen und Services zuzugreifen. ElastiCache Um eine servicebezogene Rolle für Sie zu erstellen, verwenden Sie die AWS verwaltete Richtlinie mit dem Namen `AmazonElastiCacheFullAccess`. Diese Rolle besitzt die vordefinierte Berechtigung, die der Service benötigt, um eine Service-verknüpfte Rolle für Sie zu erstellen.

Sie können sich entscheiden, anstelle der Standardrichtlinie eine benutzerseitig verwaltete Richtlinie zu verwenden. Stellen Sie in diesem Fall sicher, dass Sie entweder über Anrufberechtigungen verfügen `iam:createServiceLinkedRole` oder dass Sie die ElastiCache dienstbezogene Rolle erstellt haben.

Weitere Informationen finden Sie hier:

- [Eine neue Richtlinie erstellen](#) () IAM
- [AWS verwaltete Richtlinien für Amazon ElastiCache](#)
- [Verwenden von serviceverknüpften Rollen für Amazon ElastiCache](#)

Einrichten EC2

Sie müssen eine EC2 Instanz einrichten, von der aus Sie eine Verbindung zu Ihrem Cache herstellen.

- Wenn du noch keine EC2 Instanz hast, erfährst du hier, wie du eine EC2 Instanz einrichtest: [Erste Schritte mit EC2](#).
- Ihre EC2 Instance muss sich in derselben VPC Umgebung befinden und dieselben Sicherheitsgruppeneinstellungen wie Ihr Cache haben. Standardmäßig ElastiCache erstellt Amazon einen Cache in Ihrem Standard VPC und verwendet die Standardsicherheitsgruppe. Um diesem Tutorial zu folgen, stellen Sie sicher, dass sich Ihre EC2 Instance im Standard befindet VPC und über die Standardsicherheitsgruppe verfügt.

Gewähren Sie Netzwerkzugriff von einer VPC Amazon-Sicherheitsgruppe auf Ihren Cache

ElastiCache Selbst entworfene Cluster verwenden Port 6379 für Valkey- und OSS Redis-Befehle, und ElastiCache Serverless verwendet sowohl Port 6379 als auch Port 6380. Um erfolgreich eine Verbindung herzustellen und Valkey- oder OSS Redis-Befehle von Ihrer EC2 Instance aus auszuführen, muss Ihre Sicherheitsgruppe bei Bedarf Zugriff auf diese Ports gewähren.

ElastiCache (Memcached) verwendet die Ports 11211 und 11212, um Memcached-Befehle zu akzeptieren. Um erfolgreich eine Verbindung herzustellen und Memcached-Befehle von Ihrer EC2 Instance aus auszuführen, muss Ihre Sicherheitsgruppe den Zugriff auf diese Ports zulassen.

1. Melden Sie sich bei der an AWS Command Line Interface und öffnen Sie die [EC2Amazon-Konsole](#).
2. Wählen Sie in der Navigationsleiste unter Network & Security die Option Security Groups aus.
3. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe für Ihr Amazon ausVPC. Sofern Sie keine Sicherheitsgruppe zur ElastiCache Verwendung erstellt haben, wird diese Sicherheitsgruppe als Standard bezeichnet.
4. Klicken Sie auf die Registerkarte „Eingehend“ und führen Sie anschließend folgende Schritte aus.
 - a. Wählen Sie Bearbeiten aus.
 - b. Wählen Sie Regel hinzufügen aus.
 - c. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP Regel aus.
 - d. Wenn Sie Valkey oder Redis verwendenOSS, geben Sie in das Feld Portbereich den Text ein. 6379

Wenn Sie Memcached verwenden, geben Sie in das Feld Portbereich den Text ein. 11211
 - e. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrem Amazon starten, eine Verbindung zu Ihrem VPC Cache herstellen kann.
 - f. Wenn Sie ElastiCache Serverless verwenden, fügen Sie eine weitere Regel hinzu, indem Sie Regel hinzufügen wählen.
 - g. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP Regel aus.
 - h. Wenn Sie ElastiCache (RedisOSS) verwenden, geben 6380 Sie in das Feld Portbereich den Text ein.

Wenn Sie ElastiCache (Memcached) verwenden, geben Sie in das Feld Portbereich den Text ein. 11212

- i. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrem Amazon starten, eine Verbindung zu Ihrem VPC Cache herstellen kann.
- j. Wählen Sie Speichern.

Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein

Laden Sie das Valkey-Cli-Hilfsprogramm herunter und installieren Sie es.

Wenn Sie es ElastiCache zusammen mit Valkey verwenden, finden Sie das valkey-cli-Hilfsprogramm möglicherweise nützlich. Wenn Sie ElastiCache (RedisOSS) mit redis-cli verwenden, sollten Sie erwägen, zu valkey-cli zu wechseln, da es auch für Redis funktioniert. OSS

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her. Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2Erste Schritte“](#).
2. Laden Sie das valkey-cli Utility herunter und installieren Sie es, indem Sie den entsprechenden Befehl für Ihr Setup ausführen.

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

- Beim Installieren des redis6-Pakets wird redis6-cli mit standardmäßiger Verschlüsselungsunterstützung installiert.
- TLSBei der Installation von valkey-cli oder redis-cli ist es wichtig, Build-Unterstützung für zu haben. ElastiCache Serverless ist nur verfügbar, wenn es aktiviert ist. TLS
- Wenn der Cluster, mit dem Sie eine Verbindung herstellen, nicht verschlüsselt ist, benötigen Sie die Option Build_TLS=yes nicht.

Erstellen Sie einen serverlosen Valkey-Cache

In diesem Schritt erstellen Sie einen neuen Cache in Amazon ElastiCache.

AWS Management Console

So erstellen Sie mit der ElastiCache Konsole einen neuen Cache:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die <https://console.aws.amazon.com/connect/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey Caches aus.
3. Wählen Sie auf der rechten Seite der Konsole die Option Valkey-Cache erstellen
4. Geben Sie in den Cache-Einstellungen einen Namen ein. Sie können optional eine Beschreibung für den Cache eingeben.
5. Lassen Sie die Standardeinstellungen ausgewählt.
6. Klicken Sie auf Erstellen, um den Cache zu erstellen.
7. Sobald sich der Cache im Status "ACTIVE" befindet, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

AWS CLI

Das folgende AWS CLI Beispiel erstellt einen neuen Cache mit 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
```

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist.

Verwenden Sie den `describe-serverless-caches` Befehl, ElastiCache um zu überprüfen, ob die Erstellung des Caches abgeschlossen ist.

Linux

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

Windows

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

Fahren Sie nach dem Erstellen des neuen Caches mit [Daten in den Cache lesen und in den Cache schreiben](#) fort.

Daten in den Cache lesen und in den Cache schreiben

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

In diesem Abschnitt wird auch davon ausgegangen, dass Sie VPC Zugriff und Sicherheitsgruppeneinstellungen für die EC2 Instance eingerichtet haben, von der aus Sie eine Verbindung zu Ihrem Cache herstellen, und dass Sie Valkey-CLI auf Ihrer Instance eingerichtet haben. EC2 Weitere Informationen zu diesem Schritt finden Sie unter [einrichten ElastiCache](#).

Finden Ihres Cache-Endpunkts

AWS Management Console

So finden Sie den Endpunkt Ihres Caches mithilfe der Konsole: ElastiCache

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey-Caches aus.
3. Klicken Sie auf der rechten Seite der Konsole auf den Namen des Caches, den Sie gerade erstellt haben.
4. Suchen Sie in den Cache-Details den Cache-Endpoint und kopieren Sie ihn.

AWS CLI

Das folgende AWS CLI Beispiel zeigt, wie Sie mit dem Befehl den Endpunkt für Ihren neuen Cache finden. `describe-serverless-caches` Nachdem Sie den Befehl ausgeführt haben, suchen Sie nach dem Feld „Endpoint“.

Linux

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

Windows

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

Connect zu Ihrem Valkey Cache her (Linux)

Jetzt, da Sie den benötigten Endpunkt haben, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Der folgende Befehl stellt eine Verbindung mit einem Cache her. (Hinweis: Ersetzen Sie `cache-endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben.)

```
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 zu Ihrem Valkey Cache her (Windows)

Jetzt, da Sie den benötigten Endpunkt haben, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Mit dem folgenden Befehl stellen Sie eine Verbindung mit einem Cache her. Öffnen Sie die Befehlszeile, wechseln Sie in das OSS Verzeichnis Valkey oder Redis und führen Sie den Befehl aus (Hinweis: Ersetzen Sie `Cache_Endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben).

```
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"
```

Sie können jetzt mit [\(Optional\) Bereinigen](#) fortfahren.

(Optional) Bereinigen

Wenn Sie den von Ihnen erstellten ElastiCache Amazon-Cache nicht mehr benötigen, können Sie ihn löschen. Mit diesem Schritt wird sichergestellt, dass Ihnen keine Ressourcen in Rechnung gestellt werden, die Sie nicht nutzen. Sie können die ElastiCache Konsole, die oder die verwenden AWS CLI, ElastiCache API um Ihren Cache zu löschen.

AWS Management Console

So löschen Sie Ihren Cache mithilfe der Konsole:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey Caches aus.
3. Wählen Sie das Optionsfeld neben dem Cache, dem Sie löschen möchten.
4. Wählen Sie oben rechts Aktionen und dann Löschen aus.
5. Sie können sich optional dafür entscheiden, einen abschließenden Snapshot zu erstellen, bevor Sie Ihren Cache löschen.
6. Geben Sie auf dem Bestätigungsbildschirm Löschen den Cache-Namen erneut ein und wählen Sie die Option Löschen aus, um den Cluster zu löschen, oder Abbrechen, um den Cluster beizubehalten.

Sobald Ihr Cache diesen DELETING-Status annimmt, fallen keine Gebühren mehr für ihn an.

AWS CLI

Im folgenden AWS CLI Beispiel wird ein Cache mit dem Befehl gelöscht. `delete-serverless-cache`

Linux

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

Windows

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

Beachten Sie, dass der Wert des Felds Status auf DELETING gesetzt ist.

Sie können jetzt mit [Nächste Schritte](#) fortfahren.

Nächste Schritte

Weitere Informationen dazu ElastiCache finden Sie auf den folgenden Seiten:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache](#)
- [Protokollierung und Überwachung in Amazon ElastiCache](#)
- [ElastiCache bewährte Methoden und Caching-Strategien](#)
- [Snapshot und Wiederherstellung](#)
- [SNS-Überwachung von ElastiCache Ereignissen durch Amazon](#)

Erstellen Sie einen serverlosen Valkey- oder Redis-Cache OSS

In diesem Schritt erstellen Sie einen neuen Cache in Amazon ElastiCache.

AWS Management Console

So erstellen Sie mit der ElastiCache Konsole einen neuen Cache:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die <https://console.aws.amazon.com/connect/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey-Caches oder OSSRedis-Caches aus.
3. Wählen Sie auf der rechten Seite der Konsole „Valkey-Cache erstellen“ oder „Redis-Cache erstellen“ OSS
4. Geben Sie in den Cache-Einstellungen einen Namen ein. Sie können optional eine Beschreibung für den Cache eingeben.
5. Lassen Sie die Standardeinstellungen ausgewählt.
6. Klicken Sie auf Erstellen, um den Cache zu erstellen.
7. Sobald sich der Cache im Status "ACTIVE" befindet, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

AWS CLI

Das folgende AWS CLI Beispiel erstellt einen neuen Cache mit 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
```

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist.

Verwenden Sie den describe-serverless-caches Befehl, ElastiCache um zu überprüfen, ob die Erstellung des Caches abgeschlossen ist.

Linux

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

Windows

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

Fahren Sie nach dem Erstellen des neuen Caches mit [Daten in den Cache lesen und in den Cache schreiben](#) fort.

Daten in den Cache lesen und in den Cache schreiben

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

In diesem Abschnitt wird auch davon ausgegangen, dass Sie VPC Zugriff und Sicherheitsgruppeneinstellungen für die EC2 Instance eingerichtet haben, von der aus Sie eine Verbindung zu Ihrem Cache herstellen, und dass Sie Valkey-CLI auf Ihrer Instance eingerichtet haben. EC2 Weitere Informationen zu diesem Schritt finden Sie unter [einrichten ElastiCache](#).

Finden Ihres Cache-Endpunkts

AWS Management Console

So finden Sie den Endpunkt Ihres Caches mithilfe der Konsole: ElastiCache

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey caches OSSRedis caches aus.
3. Klicken Sie auf der rechten Seite der Konsole auf den Namen des Caches, den Sie gerade erstellt haben.
4. Suchen Sie in den Cache-Details den Cache-Endpunkt und kopieren Sie ihn.

AWS CLI

Das folgende AWS CLI Beispiel zeigt, wie Sie mit dem Befehl den Endpunkt für Ihren neuen Cache finden. `describe-serverless-caches` Nachdem Sie den Befehl ausgeführt haben, suchen Sie nach dem Feld „Endpunkt“.

Linux

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

Windows

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

Connect zu Ihrem Valkey- oder OSS Redis-Cache her (Linux)

Jetzt, da Sie den Endpunkt haben, den Sie benötigen, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Der folgende Befehl stellt eine Verbindung mit einem Cache her. (Hinweis: Ersetzen Sie `cache-endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben.)

```
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 zu Ihrem Valkey- oder OSS Redis-Cache her (Windows)

Da Sie nun den benötigten Endpunkt haben, können Sie sich bei Ihrer EC2 Instance anmelden und eine Verbindung zum Cache herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Mit dem folgenden Befehl stellen Sie eine Verbindung mit einem Cache her. Öffnen Sie die Befehlszeile, wechseln Sie zum Valkey-Verzeichnis und führen Sie den Befehl aus (Hinweis: Ersetzen Sie `Cache_Endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben).

```
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"
```

Sie können jetzt mit [\(Optional\) Bereinigen](#) fortfahren.

(Optional) Bereinigen

Wenn Sie den von Ihnen erstellten ElastiCache Amazon-Cache nicht mehr benötigen, können Sie ihn löschen. Mit diesem Schritt wird sichergestellt, dass Ihnen keine Ressourcen in Rechnung gestellt werden, die Sie nicht nutzen. Sie können die ElastiCache Konsole, die oder die verwenden AWS CLI, ElastiCache API um Ihren Cache zu löschen.

AWS Management Console

So löschen Sie Ihren Cache mithilfe der Konsole:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite der Konsole Valkey oder Redis OSS Caches aus.
3. Wählen Sie das Optionsfeld neben dem Cache, dem Sie löschen möchten.
4. Wählen Sie oben rechts Aktionen und dann Löschen aus.
5. Sie können sich optional dafür entscheiden, einen abschließenden Snapshot zu erstellen, bevor Sie Ihren Cache löschen.
6. Geben Sie auf dem Bestätigungsbildschirm Löschen den Cache-Namen erneut ein und wählen Sie die Option Löschen aus, um den Cluster zu löschen, oder Abbrechen, um den Cluster beizubehalten.

Sobald Ihr Cache diesen DELETING-Status annimmt, fallen keine Gebühren mehr für ihn an.

AWS CLI

Im folgenden AWS CLI Beispiel wird ein Cache mit dem Befehl gelöscht. `delete-serverless-cache`

Linux

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

Windows

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

```
--serverless-cache-name CacheName
```

Beachten Sie, dass der Wert des Felds Status auf DELETING gesetzt ist.

Sie können jetzt mit [Nächste Schritte](#) fortfahren.

Nächste Schritte

Weitere Informationen dazu ElastiCache finden Sie auf den folgenden Seiten:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache](#)
- [Protokollierung und Überwachung in Amazon ElastiCache](#)
- [ElastiCache bewährte Methoden und Caching-Strategien](#)
- [Snapshot und Wiederherstellung](#)
- [SNS Überwachung von ElastiCache Ereignissen durch Amazon](#)

Einen serverlosen Memcached-Cache erstellen

AWS Management Console

So erstellen Sie mit der Konsole einen neuen serverlosen Memcached-Cache: ElastiCache

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Memcached-Caches.
3. Wählen Sie auf der rechten Seite der Konsole Memcached-Cache erstellen aus.
4. Geben Sie in den Cache-Einstellungen einen Namen ein. Sie können optional eine Beschreibung für den Cache eingeben.
5. Lassen Sie die Standardeinstellungen ausgewählt.
6. Klicken Sie auf Erstellen, um den Cache zu erstellen.
7. Sobald der Cache den Status "ACTIVE" hat, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

Um einen neuen Cache mit dem zu erstellen AWS CLI

Das folgende AWS CLI Beispiel erstellt einen neuen Cache mit `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
```

Beachten Sie, dass für das Feld „Status“ der Wert `CREATING` festgelegt ist.

Verwenden Sie den `describe-serverless-caches` Befehl, ElastiCache um zu überprüfen, ob die Erstellung des Caches abgeschlossen ist.

Linux

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

Windows

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

Fahren Sie nach dem Erstellen des neuen Caches mit [Daten in den Cache lesen und in den Cache schreiben](#) fort.

Daten in den Cache lesen und in den Cache schreiben

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon-Leitfaden „EC2Erste Schritte“](#).

ElastiCache Erstellt standardmäßig einen Cache in Ihrer StandardeinstellungVPC. Stellen Sie sicher, dass Ihre EC2 Instance auch standardmäßig erstellt wurdeVPC, damit sie eine Verbindung zum Cache herstellen kann.

Finden Ihres Cache-Endpunkts

AWS Management Console

So finden Sie den Endpunkt Ihres Caches mithilfe der ElastiCache Konsole:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Memcached-Caches.
3. Klicken Sie auf der rechten Seite der Konsole auf den Namen des Caches, den Sie gerade erstellt haben.
4. Suchen Sie in den Cache-Details den Cache-Endpunkt und kopieren Sie ihn.

AWS CLI

Das folgende AWS CLI Beispiel zeigt, wie Sie mit dem `describe-serverless-caches` Befehl den Endpunkt für Ihren neuen Cache ermitteln können. Nachdem Sie den Befehl ausgeführt haben, suchen Sie nach dem Feld „Endpunkt“.

Linux

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

Windows

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

Connect mit Open her SSL

Informationen zum Herstellen einer Verbindung mithilfe von Open SSL finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#)

Herstellen einer Verbindung mithilfe des Memcached-Java-Clients

Informationen zum Herstellen einer Verbindung mithilfe des Memcached-Java-Clients finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#).

Stellen Sie mithilfe des Memcached-Clients PHP eine Verbindung her

```
<?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);
?>
```

Herstellen einer Verbindung mithilfe des Memcached-Python-Clients (Pymemcache)

Siehe memcache.readthedocs.io/en/latest/getting_started.html <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"
```

Herstellen einer Verbindung mithilfe des Memcached-NodeJS/TS-Clients (Electrode-IO Memcache)

Siehe <https://github.com/electrode-io/Memcache> und [Memcache-Client https://www.npmjs.com/package/memcache-client](https://www.npmjs.com/package/memcache-client)

Installieren über npm `npm i memcache-client`

Erstellen Sie in der Anwendung wie folgt einen TLS Memcache-Client:

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

Herstellen einer Verbindung mithilfe des Memcached-Rust-Clients (rust-memcache)

Siehe <https://crates.io/crates/memcache> und <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();
```

Herstellen einer Verbindung mithilfe des Memcached-Go-Clients (Gomemcache)

<https://github.com/bradfitz/Siehe> 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)
```

Herstellen einer Verbindung mithilfe des Memcached-Ruby-Clients (Dalli)

Siehst du 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 mit Memcached her. NETKunde () EnyimMemcachedCore

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

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

Sie können jetzt mit [\(Optional\) Bereinigen](#) fortfahren.

(Optional) Bereinigen

Mit dem AWS Management Console

Mit dem folgenden Verfahren wird ein einzelner Cache aus Ihrer Bereitstellung gelöscht. Wenn Sie mehrere Caches löschen möchten, wiederholen Sie das Verfahren für jeden Cache, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Cache fertig gelöscht ist, bevor Sie den Vorgang zum Löschen eines anderen Caches starten.

So löschen Sie einen Cache

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Konsolen-Dashboard die Engine aus, auf der der Cache, den Sie löschen möchten, läuft. Es wird eine Liste mit allen Caches, in denen diese Engine ausgeführt wird, angezeigt.
3. Wählen Sie den Namen des Caches aus der Liste der Caches aus, um den zu löschenden Cache anzugeben.

Important

Sie können jeweils nur einen Cache von der ElastiCache Konsole löschen. Werden mehrere Caches ausgewählt, wird der Löschvorgang deaktiviert.

4. Klicken Sie bei Actions auf Delete.
5. Wählen Sie auf dem Bestätigungsbildschirm Cache löschen die Option Löschen aus, um den Cache zu löschen, oder Abbrechen, wenn Sie den Cache luster beibehalten möchten.
6. Wenn Sie Löschen auswählen, ändert sich der Status des Caches in deleting.

Sobald Ihr Cache diesen DELETINGStatus annimmt, fallen keine Gebühren mehr für ihn an.

Mit dem AWS CLI

Der folgende Code löscht den Cache my-cache.

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

Die `delete-serverless-cache` CLI Aktion löscht nur einen serverlosen Cache. Um mehrere Caches zu löschen, rufen Sie `delete-serverless-cache` jeden serverlosen Cache auf, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Serverless-Cache fertig gelöscht ist, bevor Sie einen anderen löschen.

Für Linux, macOS oder Unix:

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

Für Windows:

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

Weitere Informationen finden Sie im [AWS CLI For-Thema](#). ElastiCache `delete-serverless-cache`

Sie können jetzt mit [Nächste Schritte](#) fortfahren.

Nächste Schritte

Weitere Informationen dazu finden ElastiCache Sie unter:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache](#)
- [Kontingente für ElastiCache](#)
- [ElastiCache bewährte Methoden und Caching-Strategien](#)
- [ElastiCache Ereignisse anzeigen](#)

Tutorials: Erste Schritte mit Python und ElastiCache

Dieser Abschnitt enthält praktische Tutorials, die Ihnen helfen, mehr über Valkey und ElastiCache Redis zu erfahren. OSS Wir empfehlen Ihnen, eines der sprachspezifischen Tutorials durchzuarbeiten.

Note

AWS SDKs sind für eine Vielzahl von Sprachen verfügbar. Eine vollständige Liste finden Sie unter [Tools für Amazon Web Services](#).

Themen

- [Python und ElastiCache](#)

Python und ElastiCache

In diesem Tutorial verwenden Sie AWS SDK for Python (Boto3), um einfache Programme zu schreiben, mit denen Sie die folgenden ElastiCache (Redis OSS -) Operationen ausführen können:

- Cluster ElastiCache (RedisOSS) erstellen (Clustermodus aktiviert und Clustermodus deaktiviert)
- Prüfen Sie, ob Benutzer oder Benutzergruppen existieren, andernfalls erstellen Sie sie. (Diese Funktion ist ab Valkey 7.2 und höher und ab Redis OSS 6.0 verfügbar.)
- Connect ElastiCache
- Führen Sie Aktionen wie das Festlegen und Abrufen von Strings, das Lesen von und Schreiben in Streams sowie das Veröffentlichen und Abonnieren von Pub/Sub-Kanal aus.

Während Sie dieses Tutorial durcharbeiten, können Sie sich auf die Dokumentation AWS SDK für Python (Boto) beziehen. Der folgende Abschnitt ist spezifisch für ElastiCache: [ElastiCache Low-Level-Client](#)

Tutorial-Voraussetzungen

- Richten Sie einen AWS Zugriffsschlüssel ein, um den zu verwenden. AWS SDKs Weitere Informationen finden Sie unter [einrichten ElastiCache](#).
- Python 3.0 oder später installieren. Weitere Informationen finden Sie unter <https://www.python.org/downloads>. Weiter Informationen finden Sie unter [Quickstart](#) in der Boto 3-Dokumentation.

Themen

- [Tutorial: ElastiCache Cluster und Benutzer erstellen](#)
- [Tutorial: Verbindung herstellen mit ElastiCache](#)

- [Verwendungsbeispiele](#)

Tutorial: ElastiCache Cluster und Benutzer erstellen

In den folgenden Beispielen werden boto3 SDK für ElastiCache (Redis-OSS) Verwaltungsoperationen (Cluster- oder Benutzererstellung) und redis-py-cluster redis-py/ für die Datenverarbeitung verwendet.

Themen

- [Erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus](#)
- [Erstellen Sie einen Cluster mit deaktiviertem Clustermodus mit und TLS RBAC](#)
- [Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus](#)
- [Erstellen Sie einen Cluster mit aktiviertem Clustermodus mit und TLS RBAC](#)
- [Überprüfen Sie, ob Benutzer / Benutzergruppe vorhanden sind, andernfalls erstellen Sie sie](#)

Erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens .py ein.

CreateClusterModeDisabledCluster

```
import boto3
import logging

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

def
create_cluster_mode_disabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumCacheNodes=3,
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)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python CreateClusterModeDisabledCluster.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Erstellen Sie einen Cluster mit deaktiviertem Clustermodus mit und TLS RBAC

Um die Sicherheit zu gewährleisten, können Sie Transport Layer Security (TLS) und Role-Based Access Control (RBAC) verwenden, wenn Sie einen Cluster mit deaktiviertem Clustermodus erstellen. Im Gegensatz zu Valkey oder Redis OSSAUTH, bei denen alle authentifizierten Clients vollen Zugriff auf die Replikationsgruppe haben, wenn ihr Token authentifiziert ist, RBAC können Sie den Clusterzugriff über Benutzergruppen steuern. Diese Benutzergruppen dienen dazu, den Zugriff auf Replikationsgruppen zu organisieren. Weitere Informationen finden Sie unter [Rollenbasierte Zugriffskontrolle \(\) RBAC](#).

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen `.py` ein.
ClusterModeDisabledWith RBAC

```
import boto3
import logging

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

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

    Returns a dictionary with the API response

    :param CacheNodeType: Node type used on the cluster. If not specified,
cache.t3.small will be used
    Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/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)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ClusterModeDisabledWithRBAC.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens `ClusterModeEnabled.py` ein.

```
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)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ClusterModeEnabled.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Erstellen Sie einen Cluster mit aktiviertem Clustermodus mit und TLS RBAC

Um die Sicherheit zu gewährleisten, können Sie Transport Layer Security (TLS) und Role-Based Access Control (RBAC) verwenden, wenn Sie einen Cluster mit aktiviertem Clustermodus erstellen. Im Gegensatz zu Valkey oder Redis OSSAUTH, bei denen alle authentifizierten Clients vollen Zugriff auf die Replikationsgruppe haben, wenn ihr Token authentifiziert ist, RBAC können Sie den Clusterzugriff über Benutzergruppen steuern. Diese Benutzergruppen dienen dazu, den Zugriff auf Replikationsgruppen zu organisieren. Weitere Informationen finden Sie unter [Rollenbasierte Zugriffskontrolle \(\) RBAC](#).

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen `.py` ein.
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)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ClusterModeEnabledWithRBAC.py
```

Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Überprüfen Sie, ob Benutzer / Benutzergruppe vorhanden sind, andernfalls erstellen Sie sie

Mit RBAC erstellen Sie Benutzer und weisen ihnen mithilfe einer Zugriffszeichenfolge bestimmte Berechtigungen zu. Sie weisen die Benutzer Benutzergruppen zu, denen eine bestimmte Rolle zugewiesen ist (Administratoren, Personalabteilung), die dann einer oder mehreren ElastiCache (Redis-OSS) Replikationsgruppen zugewiesen werden. Auf diese Weise können Sie Sicherheitsgrenzen zwischen Clients einrichten, die dieselbe Valkey- oder OSS Redis-Replikationsgruppe (n) verwenden, und verhindern, dass Clients gegenseitig auf die Daten zugreifen. Weitere Informationen finden Sie unter [Rollenbasierte Zugriffskontrolle \(\) RBAC](#).

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen `UserAndUserGroups .py` ein. Aktualisieren Sie den Mechanismus für die Bereitstellung von Anmeldeinformationen. In diesem Beispiel werden Anmeldeinformationen als ersetzbar angezeigt und ihnen wird ein nicht deklariertes Element zugewiesen. Vermeiden Sie die Hartkodierung von Anmeldeinformationen.

In diesem Beispiel wird eine Zugriffszeichenfolge mit den Berechtigungen für den Benutzer verwendet. Weitere Informationen zu Zugriffszeichenfolgen finden Sie unter [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#).

```
import boto3
import logging

logging.basicConfig(level=logging.INFO)
```

```
client = boto3.client('elasticache')

def check_user_exists(UserId):
    """Checks if UserId exists

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

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

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

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

def create_user(UserId=None, Username=None, Password=None, AccessString=None):
```

```
"""Creates a new user

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

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python UserAndUserGroups.py
```

Tutorial: Verbindung herstellen mit ElastiCache

In den folgenden Beispielen wird der Valkey- oder OSS Redis-Client verwendet, um eine Verbindung herzustellen. ElastiCache

Themen

- [Herstellen einer Verbindung zu einem Cluster mit deaktiviertem Cluster-Modus](#)
- [Herstellen einer Verbindung zu einem Cluster mit aktiviertem Cluster-Modus](#)

Herstellen einer Verbindung zu einem Cluster mit deaktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens ConnectClusterModeDisabled.py ein. Aktualisieren Sie den Mechanismus für die Bereitstellung von Anmeldeinformationen. In diesem Beispiel werden Anmeldeinformationen als ersetzbar angezeigt und ihnen wird ein nicht deklariertes Element zugewiesen. Vermeiden Sie die Hartkodierung von Anmeldeinformationen.

```
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")
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ConnectClusterModeDisabled.py
```

Herstellen einer Verbindung zu einem Cluster mit aktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen `ConnectClusterModeEnabled.py` ein.

```
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")
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ConnectClusterModeEnabled.py
```

Verwendungsbeispiele

In den folgenden Beispielen wird das boto3 SDK für die Arbeit mit ElastiCache (ElastiCache Redis) verwendet. OSS

Themen

- [Zeichenfolgen festlegen und abrufen](#)
- [Festlegen und Abrufen eines Hash mit mehreren Elementen](#)

- [Veröffentlichen \(Schreiben\) und Abonnieren \(Lesen\) von einem Pub/Sub-Kanal](#)
- [Schreiben und Lesen aus einem Stream](#)

Zeichenfolgen festlegen und abrufen

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens `.py` ein.

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))
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python SetAndGetStrings.py
```

Festlegen und Abrufen eines Hash mit mehreren Elementen

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens `SetAndGetHash.py` ein.

```
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))
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python SetAndGetHash.py
```

Veröffentlichen (Schreiben) und Abonnieren (Lesen) von einem Pub/Sub-Kanal

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens PubAndSub.py ein.

```
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)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python PubAndSub.py
```

Schreiben und Lesen aus einem Stream

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei namens `ReadWriteStream.py` ein.

```
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 acknowledges the message, it won't be provided again. If message
    wasn't acknowledged, it can be served to another thread.

    :param groupName: stream group where 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
                    successfully, 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)
```

Geben Sie den folgenden Befehl ein, um das Programm auszuführen:

```
python ReadWriteStream.py
```

Tutorial: Konfiguration von Lambda für den Zugriff ElastiCache in einem VPC

In diesem Tutorial erfahren Sie, wie Sie einen ElastiCache serverlosen Cache erstellen, eine Lambda-Funktion erstellen, dann die Lambda-Funktion testen und optional anschließend bereinigen.

Themen

- [Schritt 1: Einen serverlosen Cache erstellen ElastiCache .](#)
- [Schritt 2: Erstellen Sie eine Lambda-Funktion für ElastiCache](#)
- [Schritt 3: Testen Sie die Lambda-Funktion mit ElastiCache](#)
- [Schritt 4: Aufräumen \(optional\)](#)

Schritt 1: Einen serverlosen Cache erstellen ElastiCache .

Gehen Sie folgendermaßen vor, um einen serverlosen Cache zu erstellen.

Schritt 1.1: Erstellen Sie einen serverlosen Cache

In diesem Schritt erstellen Sie mit () einen serverlosen Cache im standardmäßigen Amazon VPC in der Region us-east-1 in Ihrem Konto. AWS Command Line Interface CLI Informationen zum Erstellen eines serverlosen Caches mithilfe der ElastiCache Konsole oder finden Sie unter. API [Erstellen Sie einen serverlosen Valkey-Cache](#)

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

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist. Es kann eine Minute dauern ElastiCache , bis der Cache fertig erstellt ist.

Schritt 1.2: Kopieren Sie den serverlosen Cache-Endpoint

Stellen Sie mit dem Befehl sicher, dass ElastiCache (RedisOSS) die Erstellung des Caches abgeschlossen hat. `describe-serverless-caches`

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

Kopieren Sie die in der Ausgabe angezeigte Endpunktadresse. Sie benötigen diese Adresse, wenn Sie das Bereitstellungspaket für Ihre Lambda-Funktion erstellen.

Schritt 1.3: Rolle erstellen IAM

1. Erstellen Sie für Ihre Rolle ein Dokument mit IAM Vertrauensrichtlinien, das es Ihrem Konto ermöglicht, die neue Rolle anzunehmen, wie unten dargestellt. Speichern Sie die Richtlinie in einer Datei namens `trust-policy.json`.

```
{  
  "Version": "2012-10-17",  
  "Statement": [{  
    "Effect": "Allow",  
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
    "Action": "sts:AssumeRole"  
  },  
  {  
    "Effect": "Allow",  
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },  
    "Action": "sts:AssumeRole"  
  }  
}
```

```
    "Effect": "Allow",
    "Principal": {
      "Service": "lambda.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  ]
}
```

- Erstellen Sie ein IAM Richtliniendokument, wie unten gezeigt. Speichern Sie die Richtlinie in einer Datei namens `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"
      ]
    }
  ]
}
```

- Erstellen Sie eine IAM Rolle.

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

- Erstellen Sie die IAM Richtlinie.

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

- Hängen Sie die IAM Richtlinie an die Rolle an.

```
aws iam attach-role-policy \
--role-name "elasticache-iam-auth-app" \
```

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

Schritt 1.4: Erstellen Sie einen serverlosen Cache

1. Erstellen Sie einen neuen Standardbenutzer.

```
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. Erstellen Sie einen neuen IAM -fähigen Benutzer.

```
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. Erstellen Sie eine Benutzergruppe und fügen Sie einen Benutzer an.

```
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
```

Schritt 2: Erstellen Sie eine Lambda-Funktion für ElastiCache

Gehen Sie wie folgt vor, um eine Lambda-Funktion für den Zugriff auf den ElastiCache Cache zu erstellen.

Schritt 2.1: Erstellen Sie eine Lambda-Funktion

In diesem Tutorial stellen wir Beispielcode in Python für Ihre Lambda-Funktion bereit.

Python

Der folgende Python-Beispielcode liest und schreibt ein Element in Ihren ElastiCache Cache. Kopieren Sie den Code in eine Datei und speichern Sie diese mit dem Namen `app.py`. Achten Sie darauf, den `elasticache_endpoint` Wert im Code durch die Endpunktadresse zu ersetzen, die Sie im vorherigen Schritt kopiert haben.

```
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.use1.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"
```

Dieser Code verwendet die Python-Bibliothek `redis-py`, um Elemente in Ihren Cache zu legen und abzurufen. Dieser Code verwendet `Cachetools`, um generierte IAM Auth-Token 15 Minuten lang zwischenspeichern. Gehen Sie wie folgt vor, um ein Bereitstellungspaket zu erstellen, das `Redis-py` und `Cachetools` enthält.

Erstellen Sie in Ihrem Projektverzeichnis, das die Quellcodedatei `app.py` enthält, ein Ordnerpaket, in dem Sie die Bibliotheken `redis-py` und `cachetools` installieren können.

```
mkdir package
```

Installieren Sie `redis-py`, `cachetools` mit `pip`.

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

Erstellen Sie eine ZIP-Datei, die die Bibliotheken `redis-py` und `cachetools` enthält. Führen Sie unter Linux oder macOS den folgenden Befehl aus: Verwenden Sie in Windows Ihr bevorzugtes ZIP-Hilfsprogramm, um eine `.zip`-Datei mit den Bibliotheken `redis-py` und `cachetools` im Stammverzeichnis zu erstellen.

```
cd package  
zip -r ../my_deployment_package.zip .
```

Fügen Sie den Funktionscode in die ZIP-Datei ein. Führen Sie unter Linux oder macOS den folgenden Befehl aus: Verwenden Sie in Windows Ihr bevorzugtes ZIP-Programm, um `app.py` zum Stammverzeichnis Ihrer ZIP-Datei hinzuzufügen.

```
cd ..  
zip my_deployment_package.zip app.py
```

Schritt 2.2: Erstellen Sie die IAM Rolle (Ausführungsrolle)

Hängen Sie die AWS verwaltete Richtlinie mit `AWSLambdaVPCLambdaAccessExecutionRole` dem Namen der Rolle an.

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

Schritt 2.3: Aktualisieren des Bereitstellungspakets (Erstellen der Lambda-Funktion)

In diesem Schritt erstellen Sie die Lambda-Funktion (AccessValkey) mit dem Befehl AWS CLI `create-function`.

Führen Sie in dem Projektverzeichnis, das die ZIP-Datei Ihres Bereitstellungspakets enthält, den folgenden Lambda-Befehl CLI `create-function` aus.

Verwenden Sie für die Rollenoption die ARN Ausführungsrolle, die Sie im vorherigen Schritt erstellt haben. Geben Sie für die `vpc-config` kommagetrennte Liste der Subnetze VPC Ihres Standardsystems und der Sicherheitsgruppen-ID VPC Ihres Standardserver ein. Sie finden diese Werte in der VPC Amazon-Konsole. Um die Subnetze Ihrer VPC Standardsubnetze zu finden, wählen Sie Ihre VPCs und dann die VPC Standardsubnetze Ihres AWS Kontos aus. Um die entsprechende Sicherheitsgruppe zu finden VPC, gehen Sie zu Sicherheit und wählen Sie Sicherheitsgruppen aus. Stellen Sie sicher, dass die Region `us-east-1` ausgewählt ist.

```
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
```

Schritt 3: Testen Sie die Lambda-Funktion mit ElastiCache

In diesem Schritt rufen Sie die Lambda-Funktion manuell mit dem Befehl `invoke` auf. Wenn die Lambda-Funktion ausgeführt wird, generiert sie eine UUID und schreibt sie in den ElastiCache Cache, den Sie in Ihrem Lambda-Code angegeben haben. Die Lambda-Funktion ruft das Element dann aus dem Cache ab.

1. Rufen Sie die Lambda-Funktion (AccessValkey) mit dem Befehl AWS Lambda `invoke` auf.

```
aws lambda invoke \  
--function-name AccessValkey \  
--region us-east-1 \  
output.txt
```

2. Stellen Sie sicher, dass die Lambda-Funktion erfolgreich ausgeführt wurde:

- Überprüfen Sie die Datei "output.txt".
- Überprüfen Sie die Ergebnisse in CloudWatch Logs, indem Sie die CloudWatch Konsole öffnen und die Protokollgruppe für Ihre Funktion auswählen (/aws/lambda/). AccessValkey Die Ausgabe dieses Protokollstreams sollte ähnlich wie folgt aussehen:

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

- Überprüfen Sie die Ergebnisse in der Konsole. AWS Lambda

Schritt 4: Aufräumen (optional)

Gehen Sie zum Aufräumen wie folgt vor.

Schritt 4.1: Lambda-Funktion löschen

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

Schritt 4.2: Löschen Sie den serverlosen Cache

Löschen Sie den Cache.

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

Benutzer und Benutzergruppen entfernen.

```
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
```

Schritt 4.3: IAM Rolle und Richtlinien entfernen

```
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/AWSLambdaVPCLambdaAccessExecutionRole"
```

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

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

Entwerfen und Verwalten Ihres eigenen ElastiCache Clusters

Wenn Sie eine genaue Kontrolle über Ihren ElastiCache Cluster benötigen, können Sie Ihren eigenen Cluster entwerfen. ElastiCache ermöglicht Ihnen den Betrieb eines knotenbasierten Clusters, indem Sie den Knotentyp, die Anzahl der Knoten und die Platzierung der Knoten in den Availability Zones für Ihren Cluster auswählen. AWS Da es ElastiCache sich um einen vollständig verwalteten Service handelt, verwaltet er automatisch die Hardwarebereitstellung, die Überwachung, den Austausch von Knoten und das Software-Patching für Ihren Cluster.

Weitere Informationen zum Einrichten finden Sie unter [einrichten ElastiCache](#). Details zur Verwaltung, Aktualisierung oder Löschung von Knoten oder Clustern finden Sie unter [Knoten verwalten in ElastiCache](#). Einen Überblick über die wichtigsten Komponenten einer ElastiCache Amazon-Bereitstellung beim Entwerfen Ihres eigenen ElastiCache Clusters finden Sie in diesen [Schlüsselkonzepten](#).

Themen

- [ElastiCache Komponenten und Funktionen](#)
- [ElastiCache Terminologie](#)
- [Tutorial: So entwerfen Sie Ihren eigenen Cluster](#)
- [Löschen eines Clusters](#)
- [Weitere ElastiCache Tutorials und Videos](#)
- [Knoten verwalten in ElastiCache](#)
- [Verwaltung von Clustern in ElastiCache](#)
- [Vergleich der selbst entworfenen Caches von ValkeyOSS, Redis und Memcached](#)
- [Online-Migration für Valkey oder Redis OSS](#)
- [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#)

ElastiCache Komponenten und Funktionen

Im Folgenden finden Sie einen Überblick über die wichtigsten Komponenten einer ElastiCache Amazon-Bereitstellung.

Themen

- [ElastiCache Knoten](#)
- [ElastiCache Scherben](#)
- [ElastiCache Cluster](#)
- [ElastiCache Replikation](#)
- [ElastiCache Endpunkte](#)
- [ElastiCache Parametergruppen](#)
- [ElastiCache Sicherheit](#)
- [ElastiCache Subnetzgruppen](#)
- [ElastiCache Backups](#)
- [ElastiCache Ereignisse](#)

ElastiCache Knoten

Ein Knoten ist der kleinste Baustein einer ElastiCache Bereitstellung. Ein Knoten kann isoliert von anderen Knoten oder in einer bestimmten Beziehung zu anderen Knoten existieren.

Ein Knoten ist ein sicherer, an das Netzwerk angeschlossener Block mit fester Größe. RAM Jeder Knoten führt eine Instance der Engine und der Version aus, die gewählt wurden, als Sie Ihren Cluster erstellt haben. Falls erforderlich, können Sie die Knoten in einem Cluster auf einen anderen Instance-Typ nach oben oder unten skalieren. Weitere Informationen finden Sie unter [Skalierung ElastiCache](#).

Jeder Knoten in einem Cluster ist vom selben Instance-Typ und führt dieselbe Cache-Engine aus. Jeder Cache-Knoten hat seinen eigenen Namen und Port für den Domain Name Service (DNS). Mehrere Arten von Cache-Knoten werden unterstützt, jeder mit unterschiedlich viel zugewiesenem Arbeitsspeicher. Eine Liste der unterstützten Knoten-Instance-Typen finden Sie unter [Unterstützte Knotentypen](#).

Sie können Knoten auf einer pay-as-you-go Basis erwerben, bei der Sie nur für die Nutzung eines Nodes bezahlen. Oder Sie können reservierte Knotenpunkte zu einem deutlich reduzierten Stundensatz erwerben. Wenn Ihre Nutzungsrate hoch ist, können Sie durch den Kauf reservierter Knoten Geld sparen. Angenommen, Ihr Cluster wird fast immer verwendet, und Sie fügen gelegentlich Knoten hinzu, um Spitzen abzufangen. In diesem Fall können Sie eine Reihe von reservierten Knoten erwerben, die die meiste Zeit laufen.. Sie können dann pay-as-you-go Knoten für die Zeiten kaufen, in denen Sie gelegentlich Knoten hinzufügen müssen. Weitere Informationen zu reservierten Knoten finden Sie unter [Reservierte Knoten](#).

Weitere Informationen zu Knoten finden Sie unter [Knoten verwalten in ElastiCache](#).

ElastiCache Scherben

Ein Valkey- oder OSS Redis-Shard (im API und als Knotengruppe bezeichnet CLI) ist eine Gruppierung von ein bis sechs verwandten Knoten. Ein Valkey- oder OSS Redis-Cluster mit aktiviertem Clustermodus hat immer mindestens einen Shard.

Sharding ist eine Methode der Datenbankpartitionierung, die große Datenbanken in kleinere, schnellere und einfacher zu verwaltende Teile unterteilt, die als Daten-Shards bezeichnet werden. Dadurch kann die Datenbankeffizienz erhöht werden, indem Operationen auf mehrere separate Abschnitte verteilt werden. Die Verwendung von Shards kann viele Vorteile bieten, darunter eine verbesserte Leistung, Skalierbarkeit und Kosteneffizienz.

Valkey- und OSS Redis-Cluster mit aktiviertem Clustermodus können bis zu 500 Shards enthalten, wobei Ihre Daten auf die Shards verteilt sind. Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Valkey- oder OSS Redis-Engine-Version 5.0.6 oder höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#). Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, siehe [AWS -Service-Limits](#) und wählen Sie den Limittyp Knoten pro Cluster pro Instance-Typ.

Ein Shard mit mehreren Knoten implementiert die Replikation durch einen Primärknoten mit Lese-/Schreibzugriff und 1-5 Replikationsknoten. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Weitere Informationen zu Shards finden Sie unter [Arbeiten mit Shards in ElastiCache](#).

ElastiCache Cluster

Ein Cluster ist eine logische Gruppierung von einem oder mehreren [Knoten](#). Daten werden auf die Knoten in einem Memcache-Cluster und auf die Shards in einem Valkey- oder OSS Redis-Cluster, für den der Clustermodus aktiviert ist, partitioniert.

Viele Operationen sind ElastiCache auf Cluster ausgerichtet:

- Erstellen eines Clusters
- Modifizieren eines Clusters
- Erstellen von Snapshots eines Clusters (alle Redis-Versionen)
- Löschen eines Clusters
- Anzeigen der Elemente in einem Cluster
- Hinzufügen oder Entfernen von Kostenzuordnungs-Tags in einem Cluster

Detailliertere Informationen finden Sie in den folgenden verwandten Themen:

- [Verwaltung von Clustern in ElastiCache](#) und [Knoten verwalten in ElastiCache](#)

Informationen zu Clustern, Knoten und verwandten Operationen

- [AWS Servicebeschränkungen: Amazon ElastiCache](#)

Informationen zu ElastiCache Grenzwerten, wie z. B. der maximalen Anzahl von Knoten oder Clustern. Um bestimmte dieser Grenzwerte zu überschreiten, können Sie über das [Antragsformular für ElastiCache Amazon-Cache-Knoten eine Anfrage stellen](#).

- [Minimieren von Ausfällen](#)

Informationen zur Verbesserung der Fehlertoleranz Ihrer Cluster und Valkey- oder OSS Redis-Replikationsgruppen.

Typische Cluster-Konfigurationen

Folgende sind typische Clusterkonfigurationen.

Valkey- oder Redis-Cluster OSS

Valkey- oder OSS Redis-Cluster mit deaktiviertem Clustermodus enthalten immer nur einen Shard (in der Gruppe API und CLI, eine Knotengruppe). Ein Valkey- oder OSS Redis-Shard enthält ein bis sechs Knoten. Wenn sich mehr als ein Knoten in einer Shard befindet, unterstützt die Shard die Replikation. Dabei ist ein Knoten der Primärknoten mit Lese/Schreibzugriff und die anderen sind nur lesende Replikationsknoten.

Für eine verbesserte Fehlertoleranz empfehlen wir, mindestens zwei Knoten in einem Valkey- oder OSS Redis-Cluster zu haben und Multi-AZ zu aktivieren. Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Wenn sich die Nachfrage nach Ihrem Valkey- oder OSS Redis-Cluster ändert, können Sie nach oben oder unten skalieren. Verschieben Sie dazu Ihren Cluster auf einen anderen Knoteninstanztyp. Wenn Ihre Anwendung leseintensiv ist, empfehlen wir, dem Cluster schreibgeschützte Replikate hinzuzufügen. So können Sie die Lesevorgänge auf eine angemessenere Anzahl von Knoten verteilen.

Sie können auch Daten-Tiering verwenden. Daten, auf die häufiger zugegriffen wird, werden im Speicher gespeichert, und Daten, auf die seltener zugegriffen wird, werden auf der Festplatte gespeichert. Der Vorteil der Verwendung von Daten-Tiering besteht darin, dass der Speicherbedarf verringert wird. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

ElastiCache unterstützt das dynamische Ändern des Knotentyps eines Valkey- oder OSS Redis-Clusters in einen größeren Knotentyp. Information zur Skalierung nach oben oder unten finden Sie unter [Skalierung von Einzelknotenclustern für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#) oder [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

Typische Cluster-Konfigurationen für Memcached

Memcached unterstützt bis zu 300 Knoten pro Kunde für jede AWS Region, wobei jeder Cluster aus 1—60 Knoten besteht. In einem Memcached-Cluster können Sie die Daten über die Knoten partitionieren.

Wenn Sie die Memcached-Engine ausführen, können Cluster aus 1—60 Knoten bestehen. Sie partitionieren die Datenbank über die Knoten. Ihre Anwendung liest und schreibt auf den Endpunkt eines jeden Knotens. Weitere Informationen finden Sie unter [Auto Discovery](#).

Um die Fehlertoleranz zu verbessern, platzieren Sie Ihre Memcached-Knoten in verschiedenen Availability Zones (AZs) innerhalb der Clusterregion. AWS Dadurch wirkt sich ein Fehler in einer AZ nur minimal auf den gesamten Cluster und die Anwendung aus. Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Wenn sich die Anforderungen an den Memcached-Cluster ändern, können Sie ihn durch Hinzufügen oder Entfernen von Knoten skalieren. Dabei werden die Daten entsprechend der neuen Anzahl an Knoten neu partitioniert. Zur Partitionierung der Daten empfehlen wir, konsistentes Hashing zu verwenden. Weitere Informationen zu konsistentem Hashing finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

ElastiCache Replikation

Für Valkey und Redis wird die Replikation implementiert OSS, indem zwei bis sechs Knoten in einem Shard gruppiert werden (im API und CLI, was als Knotengruppe bezeichnet wird). Einer dieser Knoten ist der primäre Knoten für Lese- und Schreibvorgänge. Alle anderen Knoten sind schreibgeschützte Replikationsknoten. Replikationen sind nur für ElastiCache Valkey und Redis OSS verfügbar und nicht für (Memcached). ElastiCache

Jeder Replikationsknoten speichert eine Kopie der Daten vom primären Knoten. Replikationsknoten verwenden asynchrone Replikationsmechanismen, um die Synchronisierung mit dem primären Knoten aufrecht zu erhalten. Anwendungen können von jedem Knoten im Cluster lesen, aber nur auf primäre Knoten schreiben. Read Replicas erhöhen die Skalierbarkeit, indem sie Lesevorgänge über mehrere Endpunkte verteilen. Read Replicas verbessern auch die Fehlertoleranz, indem sie mehrere Kopien der Daten speichern. Durch die Verteilung von Read Replicas auf mehrere Availability Zones wird die Fehlertoleranz noch weiter verbessert. Weitere Informationen zur Fehlertoleranz finden Sie unter [Minimieren von Ausfällen](#).

Valkey- oder OSS Redis-Cluster unterstützen einen Shard (im und, eine so genannte Knotengruppe).
API CLI

Bei der Replikation aus CLI Sicht von API und aus wird eine andere Terminologie verwendet, um die Kompatibilität mit früheren Versionen zu gewährleisten, aber die Ergebnisse sind dieselben. In der folgenden Tabelle sind die CLI Bedingungen API und Bedingungen für die Implementierung der Replikation aufgeführt.

Vergleich der Replikation: Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert) --> Valkey- oder Redis-Cluster mit aktiviertem Clustermodus im Vergleich zu Valkey- oder OSS Redis-Cluster mit deaktiviertem Clustermodus OSS

In der folgenden Tabelle finden Sie einen Vergleich der Funktionen der Replikationsgruppen Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis (Clustermodus aktiviert). OSS

	Valkey- oder OSS Redis-Cluster mit deaktiviertem Clustermodus	Valkey- oder OSS Redis-Cluster mit aktiviertem Clustermodus
Shards (Knotengruppen)	1	1–500

	Valkey- oder OSS Redis-Cluster mit deaktiviertem Clustermodus	Valkey- oder OSS Redis-Cluster mit aktiviertem Clustermodus
Replikate pro Shard (Knotengruppe)	0–5	0–5
Datenpartitionierung	Nein	Ja
Hinzufügen/Löschen von Replikaten	Ja	Ja
Hinzufügen/Löschen von Knotengruppen	Nein	Ja
Unterstützung einer Erweiterung	Ja	Ja
Unterstützung von Engine-Updates	Ja	Ja
Hochstufen eines Replikats auf Primär	Ja	Automatisch
Multi-AZ	Optional	Erforderlich
Sicherung/Wiederherstellung	Ja	Ja

Hinweise:

Wenn sich auf einem Primärsystem keine Replikate befinden, verlieren Sie bei einem Ausfall des Primärsystems alle darauf befindlichen Daten.

Sie können Backup und Restore verwenden, um zu Valkey oder Redis zu migrieren OSS (Clustermodus aktiviert).

Sie können Backup und Restore verwenden, um die Größe Ihres Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) zu ändern.

Alle Shards (in den Knotengruppen API und CLI,) und Knoten müssen sich in derselben Region befinden. AWS Sie können die einzelnen Knoten jedoch in mehreren Availability Zones innerhalb dieser AWS Region bereitstellen.

Lesereplikate schützen vor möglichem Datenverlust, da Ihre Daten über zwei oder mehr Knoten repliziert werden – den primären und einen oder mehrere Lesereplikate. Für eine höhere Zuverlässigkeit und schnellere Wiederherstellung empfehlen wir, ein oder mehrere Lesereplikate in verschiedenen Availability Zones zu erstellen.

Sie können auch globale Datenspeicher nutzen. Mithilfe der OSS Funktion Global Datastore for Redis können Sie mit einer vollständig verwalteten, schnellen, zuverlässigen und sicheren Replikation über Regionen hinweg arbeiten. AWS Mit dieser Funktion können Sie regionsübergreifende Read Replica-Cluster erstellen, um Lesevorgänge mit niedriger Latenz und regionsübergreifende Disaster Recovery ElastiCache zu ermöglichen. AWS Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe](#) globaler Datenspeicher.

Replikation: Einschränkungen und Ausnahmen

- Multi-AZ wird für den Knotentypen T1 nicht unterstützt.

ElastiCache Endpunkte

Ein Endpunkt ist die eindeutige Adresse, die Ihre Anwendung verwendet, um eine Verbindung zu einem ElastiCache Knoten oder Cluster herzustellen.

Einzelknoten-Endpunkte für Valkey oder Redis OSS mit deaktiviertem Clustermodus

Der Endpunkt für einen Valkey- oder OSS Redis-Cluster mit einem Knoten wird verwendet, um eine Verbindung zum Cluster sowohl für Lese- als auch für Schreibvorgänge herzustellen.

Endpunkte mit mehreren Knoten für Valkey oder Redis mit deaktiviertem Clustermodus OSS

Ein Valkey- oder OSS Redis-Cluster mit mehreren Knoten und deaktiviertem Clustermodus hat zwei Arten von Endpunkten. Über den primären Endpunkt wird immer die Verbindung zum primären Knoten im Cluster hergestellt, auch wenn sich die primäre Rolle des Knotens ändert. Verwenden Sie den primären Endpunkt für alle Schreibvorgänge im Cluster.

Verwenden Sie Leser-Endpoint, um am Endpoint ankommende Verbindungen auf alle Lesereplikate zu verteilen. Verwenden Sie die einzelnen Node-Endpoints für Lesevorgänge (im FeldAPI/werden CLI diese als Lese-Endpoints bezeichnet).

Valkey- oder Redis-Endpoints OSS (Clustermodus aktiviert)

Ein Valkey- oder OSS Redis-Cluster mit aktiviertem Clustermodus hat einen einzigen Konfigurationsendpoint. Über die Verbindung zum Konfigurationsendpoint findet die Anwendung für jede Shard im Cluster die primären Endpunkte sowie jene für Schreibvorgänge.

Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

ElastiCache (Memcached) Endpunkte

Jeder Knoten in einem Memcached-Cluster hat einen eigenen Endpoint. Der Cluster verfügt auch über einen sogenannten Konfigurationsendpoint. Wenn Sie Auto Discovery aktivieren und eine Verbindung zum Konfigurationsendpoint herstellen, kennt die Anwendung jeden Knotenendpoint, auch wenn danach Knoten im Cluster hinzugefügt oder entfernt wurden. Weitere Informationen finden Sie unter [Auto Discovery](#).

Weitere Informationen finden Sie unter [-Endpunkte](#).

ElastiCache Parametergruppen

Cache-Parametergruppen vereinfachen die Verwaltung von Laufzeiteinstellungen für unterstützte Engine-Software. Parameter werden zur Steuerung von Speichernutzung, Bereinigungsrichtlinien, Elementgrößen usw verwendet. Eine ElastiCache Parametergruppe ist eine benannte Sammlung von Engine-spezifischen Parametern, die Sie auf einen Cluster anwenden können. Dadurch stellen Sie sicher, dass alle Knoten in diesem Cluster identisch konfiguriert werden.

Eine Liste der unterstützten Parameter, ihrer Standardwerte und der Parameter, die geändert werden können, finden Sie unter [DescribeEngineDefaultParameters](#) (CLI: [describe-engine-default-parameters](#)).

Ausführlichere Informationen zu ElastiCache Parametergruppen finden Sie unter [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#).

ElastiCache Sicherheit

Aus Sicherheitsgründen ist der ElastiCache Knotenzugriff auf Anwendungen beschränkt, die auf den von Ihnen zugelassenen EC2 Amazon-Instances ausgeführt werden. Sie können die EC2 Amazon-Instances, die auf Ihren Cluster zugreifen können, mithilfe von Sicherheitsgruppen steuern.

Standardmäßig werden alle neuen ElastiCache Cluster in einer Amazon Virtual Private Cloud (AmazonVPC) -Umgebung gestartet. Sie können Subnetzgruppen verwenden, um Cluster-Zugriff von EC2 Amazon-Instances aus zu gewähren, die in bestimmten Subnetzen ausgeführt werden.

Zusätzlich zur Beschränkung des Knotenzugriffs ElastiCache unterstützt TLS es die direkte Verschlüsselung von Knoten, auf denen bestimmte Versionen von ausgeführt werden. ElastiCache Weitere Informationen finden Sie hier:

- [Datensicherheit bei Amazon ElastiCache](#)
- [Authentifizierung mit den Befehlen Valkey und Redis OSS AUTH](#)

ElastiCache Subnetzgruppen

Eine Subnetzgruppe ist eine Sammlung von Subnetzen (normalerweise privat), die Sie für Ihre Cluster festlegen können, die in einer Amazon-Umgebung ausgeführt werden. VPC

Wenn Sie einen Cluster in einem Amazon erstellenVPC, müssen Sie eine Cache-Subnetzgruppe angeben. ElastiCache verwendet diese Cache-Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Cache-Knoten zugeordnet werden sollen.

Weitere Informationen zur Nutzung von Cache-Subnetzgruppen in einer VPC Amazon-Umgebung finden Sie im Folgenden:

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#)
- [Subnetze und Subnetzgruppen](#)

ElastiCache Backups

Ein Backup ist eine point-in-time Kopie eines Valkey- oder OSS Redis-Clusters oder eines serverlosen Caches oder eines serverlosen Memcached-Caches. Backups können zur

Wiederherstellung eines bestehenden Clusters oder zum Seeding eines neuen Clusters verwendet werden. Sicherungen umfassen alle Daten in einem Cluster sowie zusätzlich einige Metadaten.

Abhängig von der Version von Valkey oder Redis, die auf Ihrem Cluster OSS ausgeführt wird, benötigt der Backup-Prozess unterschiedliche Mengen an reserviertem Speicher, um erfolgreich zu sein. Weitere Informationen finden Sie hier:

- [Snapshot und Wiederherstellung](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Auswirkungen von Backups selbst entworfener Cluster auf die Leistung](#)
- [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#)

ElastiCache Ereignisse

ElastiCache Sendet bei wichtigen Ereignissen in einem Cache-Cluster eine Benachrichtigung an ein bestimmtes SNS Amazon-Thema. Zu diesen Ereignissen gehören z. B. das fehlgeschlagene oder erfolgreiche Hinzufügen eines Knotens, die Änderung einer Sicherheitsgruppe usw. Durch die Überwachung von Schlüsselereignissen können Sie den aktuellen Zustand Ihrer Cluster erkennen und in vielen Fällen Korrekturmaßnahmen ergreifen.

Weitere Informationen zu ElastiCache Ereignissen finden Sie unter [SNSÜberwachung von ElastiCache Ereignissen durch Amazon](#).

ElastiCache Terminologie

Im Oktober 2016 ElastiCache startete Amazon die Unterstützung für Redis OSS 3.2. Zu diesem Zeitpunkt haben wir Unterstützung für die Partitionierung Ihrer Daten auf bis zu 500 Shards (im und als Knotengruppen bezeichnet) hinzugefügt. ElastiCache API AWS CLI Um die Kompatibilität mit früheren Versionen zu gewährleisten, haben wir den Betrieb von API Version 2015-02-02 um die neue Redis-Funktionalität erweitert. OSS

Gleichzeitig haben wir begonnen, in der ElastiCache Konsole eine Terminologie zu verwenden, die in dieser neuen Funktion verwendet wird und in der gesamten Branche üblich ist. Diese Änderungen bedeuten, dass sich die in der API und verwendete Terminologie an einigen Stellen von der in der Konsole verwendeten Terminologie unterscheiden CLI kann. In der folgenden Liste sind Begriffe aufgeführt, die sich zwischen API und CLI und der Konsole unterscheiden können.

Cache-Cluster oder Knoten vs. Knoten

Es besteht eine one-to-one Beziehung zwischen einem Knoten und einem Cache-Cluster, wenn keine Replikatknoten vorhanden sind. Daher wurden die Begriffe in der ElastiCache Konsole häufig synonym verwendet. Die Konsole verwendet jetzt durchgängig den Begriff Knoten. Einzige Ausnahme ist die Schaltfläche Create Cluster, über die der Prozess zur Erstellung eines Clusters mit oder ohne Replikationsknoten gestartet wird.

Sie ElastiCache API und verwenden die Begriffe AWS CLI weiterhin wie in der Vergangenheit.

Cluster im Vergleich zu Valkey- oder OSS Redis-Replikationsgruppe

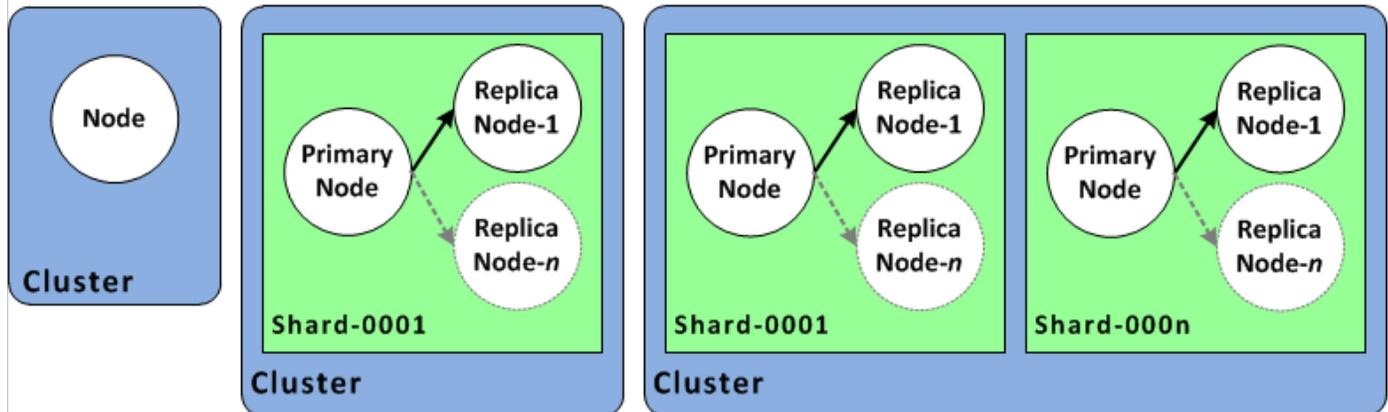
Die Konsole verwendet jetzt den Begriff Cluster für alle ElastiCache (Redis OSS -) Cluster. Die Konsole verwendet den Begriff "Cluster" in den folgenden Fällen:

- Wenn der Cluster ein Valkey- oder OSS Redis-Cluster mit einem Knoten ist.
- Wenn es sich bei dem Cluster um einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) handelt, der die Replikation innerhalb eines einzelnen Shards (im API und CLI, einer sogenannten Knotengruppe) unterstützt.
- Wenn es sich bei dem Cluster um einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) handelt, der die Replikation innerhalb von 1—90 Shards oder bis zu 500 Shards mit einer Anforderung zur Erhöhung des Limits unterstützt. Um eine Erhöhung des Limits zu beantragen, siehe [AWS -Service-Limits](#) nach und wählen Sie den Limittyp Knoten pro Cluster pro Instance-Typ.

Weitere Informationen zu Valkey- oder OSS Redis-Replikationsgruppen finden Sie unter. [Hohe Verfügbarkeit mit Replikationsgruppen](#)

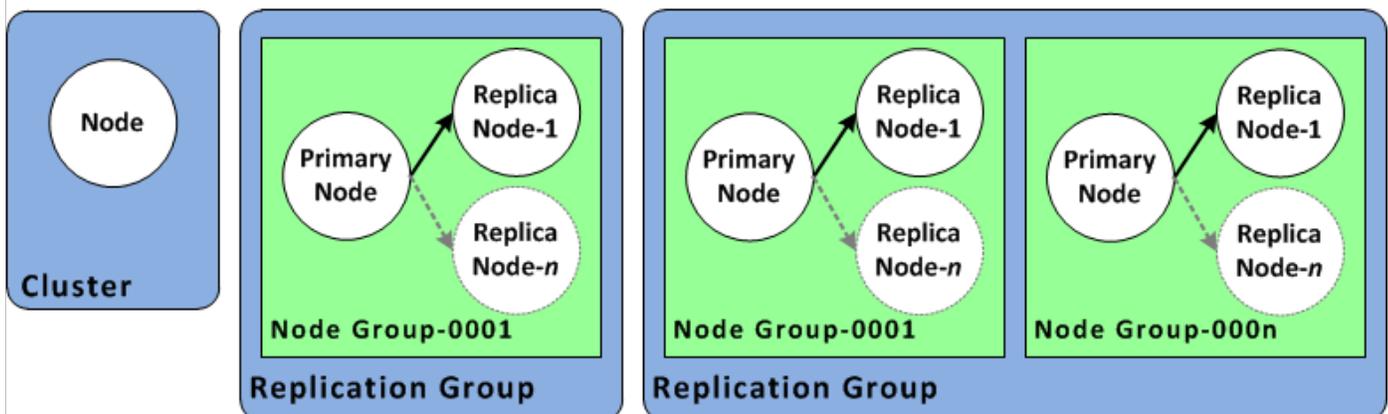
Das folgende Diagramm veranschaulicht die verschiedenen Topologien von ElastiCache (Redis-OSS) Clustern aus Sicht der Konsole.

ElastiCache (Redis OSS): Console View



Bei den AWS CLI Operationen ElastiCache API wird immer noch zwischen Einzelknotenclustern ElastiCache (RedisOSS) und Valkey- oder OSS Redis-Replikationsgruppen mit mehreren Knoten unterschieden. Das folgende Diagramm veranschaulicht die verschiedenen ElastiCache (Redis-OSS) Topologien aus der AND-Perspektive. ElastiCache API AWS CLI

ElastiCache (Redis OSS): API/CLI View



Valkey- oder OSS Redis-Replikationsgruppe im Vergleich zum globalen Datenspeicher

Ein globaler Datenspeicher ist eine Sammlung von einem oder mehreren Clustern, die sich regionsübergreifend aufeinander replizieren, wohingegen eine Valkey- oder OSS Redis-Replikationsgruppe Daten über einen Cluster mit aktiviertem Clustermodus mit mehreren Shards repliziert. Ein globaler Datenspeicher besteht aus folgenden Komponenten:

- Primärer (aktiver) Cluster – Ein primärer Cluster nimmt Schreibvorgänge entgegen, die auf alle Cluster innerhalb des globalen Datenspeichers repliziert werden. Ein primärer Cluster akzeptiert auch Leseanforderungen.
- Sekundärer (passiver) Cluster – Ein sekundärer Cluster nimmt nur Leseanforderungen entgegen und repliziert Datenaktualisierungen von einem primären Cluster. Ein sekundärer Cluster muss sich in einer anderen Region befinden als der primäre Cluster. AWS

Informationen zu globalen Datenspeichern finden Sie unter [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#).

Tutorial: So entwerfen Sie Ihren eigenen Cluster

Hier erfahren Sie, wie Sie Ihren eigenen Cluster für Valkey und OSS Redis entwerfen.

Themen

- [Entwerfen Sie Ihren eigenen ElastiCache \(Valkey-\) Cluster](#)
- [Entwerfen Sie Ihren eigenen ElastiCache \(Redis-OSS\) Cluster](#)

Entwerfen Sie Ihren eigenen ElastiCache (Valkey-) Cluster

Im Folgenden finden Sie die einmaligen Maßnahmen, die Sie ergreifen müssen, um mit dem Entwerfen Ihres ElastiCache (Valkey-) Clusters zu beginnen.

Schritt 1: Erstellung einer Subnetzgruppe

Bevor Sie einen ElastiCache (Valkey-) Cluster erstellen, erstellen Sie zunächst eine Subnetzgruppe. Eine Cache-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie möglicherweise für Ihre Cache-Cluster in einer angeben möchten. VPC Wenn Sie einen Cache-Cluster in einem startenVPC, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache Verwendet dann diese Cache-Subnetzgruppe, um jedem Cache-Knoten im Cluster IP-Adressen innerhalb dieses Subnetzes zuzuweisen.

Wenn Sie eine neue Subnetzgruppe erstellen, notieren Sie sich die Anzahl der verfügbaren IP-Adressen. Wenn das Subnetz nur über wenige freie IP-Adressen verfügt, beschränkt dies auch die Anzahl der neuen Knoten, die Sie zu dem Cluster hinzufügen können. Um dieses Problem zu lösen, können Sie einer Subnetzgruppe weitere Subnetze zuweisen, um ausreichend IP-Adressen in der

Availability Zone Ihres Clusters bereitzustellen. Danach können Sie dem Cluster weitere Knoten hinzufügen.

Weitere Informationen zur Einrichtung ElastiCache finden Sie unter [einrichten ElastiCache](#)

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Subnetzgruppe mit dem Namen mysubnetgroup (Konsole) und die AWS CLI erstellen.

Erstellen einer Subnetzgruppe (Konsole)

Im folgenden Verfahren wird das Erstellen einer Subnetzgruppe (Konsole) erläutert.

Erstellen einer DB-Sicherheitsgruppe (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie in der Navigationsliste auf Subnet Groups.
3. Klicken Sie auf Create Subnet Group (Subnetzgruppe ändern).
4. Gehen Sie im Assistenten Create Subnet Group wie folgt vor. Wenn Sie alle gewünschten Einstellungen vorgenommen haben, klicken Sie auf Yes, Create.
 - a. Geben Sie im Feld Name einen Namen für Ihre Subnetzgruppe ein.
 - b. Geben Sie im Feld Description eine Beschreibung für Ihre Subnetzgruppe ein.
 - c. Wählen Sie im Feld VPCID den Amazon VPC aus, den Sie erstellt haben.
 - d. Wählen Sie in den Listen Availability Zone und Subnet ID die Availability Zone oder [Verwenden von lokalen Zonen mit ElastiCache](#) die ID Ihres privaten Subnetzes aus und klicken Sie dann auf Hinzufügen.

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-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. Klicken Sie in der angezeigten Bestätigungsmeldung auf Close.

Ihre neue Subnetzgruppe wird in der Liste der Subnetzgruppen der Konsole angezeigt. ElastiCache Unten im Fenster können Sie die Subnetzgruppe auswählen, um Details wie die der Gruppe zugeordneten Subnetze anzuzeigen.

Erstellen einer Subnetzgruppe (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `create-cache-subnet-group` ein, um eine Subnetzgruppe zu erstellen.

Für Linux, macOS oder Unix:

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

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

Für Windows:

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

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

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

Weitere Informationen finden Sie im Thema AWS CLI [create-cache-subnet-group](#).

Schritt 2: Erstellen eines Clusters

Bevor Sie einen Cluster für die Produktion erstellen, müssen Sie natürlich überlegen, wie Sie den Cluster entsprechend Ihren geschäftlichen Anforderungen konfigurieren. Diese Probleme werden im [Einen Cluster vorbereiten in ElastiCache](#)-Abschnitt adressiert. Für die Zwecke dieser Übung „Erste Schritte“ erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus. Sie können die Standardkonfigurationswerte dort akzeptieren, wo sie zutreffen.

Der Cluster, den Sie gleich starten werden, wird live sein und nicht in einer Sandbox ausgeführt. Es fallen die ElastiCache Standardnutzungsgebühren für die Instance an, bis Sie sie löschen. Die Gesamtkosten werden minimal sein (meistens geringer als ein Dollar), wenn Sie diese Übung in einer

Sitzung durchlaufen und den Cluster löschen, sobald Sie die Übung abgeschlossen haben. Weitere Informationen zu den ElastiCache Nutzungsraten finden Sie auf [Amazon ElastiCache](#).

Ihr Cluster wird in einer virtuellen privaten Cloud (VPC) gestartet, die auf dem VPC Amazon-Service basiert.

Erstellen eines Valkey-Clusters (Cluster-Modus deaktiviert) (Konsole)

So erstellen Sie mit der Konsole einen Valkey-Cluster (Clustermodus deaktiviert) ElastiCache

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der Sie diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Erstellen VPC und folgen Sie den unter [Erstellen einer virtuellen privaten Cloud \(\) VPC](#) beschriebenen Schritten.
5. Wählen Sie auf der ElastiCache Dashboard-Seite Valkey-Cache oder OSSRedis-Cache und dann Valkey-Cache erstellen oder Redis-Cache aus. OSS
6. Führen Sie unter Cluster settings (Cluster-Einstellungen) die folgenden Schritte aus:
 - a. Wählen Sie Configure and create a new cluster (Neuen Cluster konfigurieren und erstellen) aus.
 - b. Wählen Sie für Cluster mode (Cluster-Modus) Disabled (Deaktiviert) aus.
 - c. Geben Sie für Cluster info (Cluster-Info) einen Wert für Name (Name) ein.
 - d. (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
7. Unter Location (Speicherort):

AWS Cloud

1. Wir empfehlen für AWS Cloud die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache \(Redis\) mit Multi-AZ](#). OSS
2. Unter Cluster settings (Cluster-Einstellungen):
 - a. Wählen Sie unter Engine version (Engine-Version) eine verfügbare Version aus.

- b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.
- c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis-Parameter OSS](#) und [Eine ElastiCache Parametergruppe erstellen](#).

 Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parteigruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼)).
Wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

- e. Wählen Sie unter Number of replicas (Anzahl der Repliken) die für diesen Cluster gewünschte Anzahl von Read Replicas aus. Wenn Multi-AZ aktiviert ist, muss die Zahl zwischen 1–5 liegen.
3. Unter Connectivity (Konnektivität)
 - a. Wählen Sie als Network type (Netzwerktyp) die IP-Version(en) aus, die dieser Cluster unterstützen soll.
 - b. Wählen Sie für Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten

zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only. IPv6

Geben Sie beim Erstellen einer neuen Subnetzgruppe die ID ein, zu der sie gehört. VPC

Weitere Informationen finden Sie unter:

- [Auswahl eines Netzwerktyps in ElastiCache.](#)
- [Erstellen Sie ein Subnetz in Ihrem. VPC](#)

Wenn Sie [Verwenden von lokalen Zonen mit ElastiCache](#) sind, müssen Sie ein Subnetz erstellen oder auswählen, das in der lokalen Zone liegt.

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen.](#)

4. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:

- Keine Präferenz — ElastiCache wählt die Availability Zone.
- Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache.](#)

5. Wählen Sie Next (Weiter).

6. Unter den erweiterten Valkey- oder Redis-Einstellungen OSS

- Für Security (Sicherheit):
 - i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:
 - Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand.](#)

Note

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie vom Kunden verwalteter AWS KMS Schlüssel und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Vom Kunden verwaltete Schlüssel verwenden von AWS KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey und für Redis OSS Engine Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
 - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
 - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Benutzergruppen mit der Konsole verwalten und CLI](#).
 - AUTHStandardbenutzer — Ein Authentifizierungsmechanismus für den Redis-Server. OSS Weitere Informationen finden Sie unter [AUTH](#).
- AUTH— Ein Authentifizierungsmechanismus für den OSS Redis-Server. Weitere Informationen finden Sie unter [AUTH](#).

Note

Für Valkey- und OSS Redis-Versionen ab 3.2.6, mit Ausnahme von Version 3.2.10, ist Redis die einzige Option. OSS AUTH

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre verwenden oder eine neue erstellen. VPC

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Sie VPC](#) im VPCAmazon-Benutzerhandbuch.

7. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zur OSS Sicherung und Wiederherstellung von Redis finden Sie unter [Snapshot und Wiederherstellung](#).

8. (Optional) Geben Sie ein Wartungsfenster an. Das Wartungsfenster ist die Zeit, in der Regel eine Stunde, jede Woche, zu der die ElastiCache Systemwartung für Ihren Cluster geplant wird. Sie können ElastiCache den Tag und die Uhrzeit für Ihr Wartungsfenster wählen lassen (keine Präferenz), oder Sie können Tag, Uhrzeit und Dauer selbst wählen (Wartungsfenster angeben). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Zeiten sind UCT Zeiten.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

9. (Optional) Für Protokolle:
 - Wählen Sie unter Protokollformat entweder Text oder aus JSON.
 - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppenamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppenamen oder Ihren Firehose-Streamnamen aus.
10. Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Vorschlagworten Sie Ihre Ressourcen ElastiCache](#).
11. Wählen Sie Weiter.
12. Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Sobald Sie bereit sind, klicken Sie auf Create (Erstellen).

On premises

1. Für On premises (On-Premises) empfehlen wir Ihnen, Auto-failover (Automatisches Failover) aktiviert zu lassen. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache \(RedisOSS\)](#) mit Multi-AZ
2. Um die Erstellung des Clusters abzuschließen, befolgen Sie die Schritte unter [Verwenden von Outposts](#).

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Valkey-Clusters (Cluster-Modus deaktiviert) (AWS CLI)

Example

Der folgende CLI Code erstellt einen Valkey-Cache-Cluster (Clustermodus deaktiviert) ohne Replikate.

Für Linux, macOS oder 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
```

Für 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
```

Informationen zum Arbeiten mit aktiviertem Cluster-Modus finden Sie in den folgenden Themen:

- Wenn Sie die Konsole verwenden möchten, lesen Sie [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#).
- Informationen zur Verwendung von finden Sie unter AWS CLI. [Erstellen eines Valkey- oder Redis-Clusters OSS \(Clustermodus aktiviert\) \(\)AWS CLI](#)

Schritt 3. Autorisieren Sie den Zugriff auf den Cluster

In diesem Abschnitt wird davon ausgegangen, dass Sie mit dem Starten und Herstellen einer Verbindung zu EC2 Amazon-Instances vertraut sind. Weitere Informationen finden Sie im [Amazon EC2 Getting Started Guide](#).

Alle ElastiCache Cluster sind so konzipiert, dass sie von einer EC2 Amazon-Instance aus aufgerufen werden können. Das gängigste Szenario ist der Zugriff auf einen ElastiCache Cluster von einer EC2 Amazon-Instance in derselben Amazon Virtual Private Cloud (AmazonVPC), was bei dieser Übung der Fall sein wird.

Standardmäßig ist der Netzwerkzugriff auf Ihren Cluster auf das Konto beschränkt, über das er erstellt wurde. Bevor Sie von einer EC2 Instance aus eine Verbindung zu einem Cluster herstellen können, müssen Sie die EC2 Instance für den Zugriff auf den Cluster autorisieren.

Der häufigste Anwendungsfall ist, wenn eine auf einer EC2 Instance bereitgestellte Anwendung eine Verbindung zu einem Cluster in derselben VPC Instanz herstellen muss. Der einfachste Weg, den Zugriff zwischen EC2 Instanzen und Clustern auf derselben Ebene zu verwalten, VPC besteht darin, wie folgt vorzugehen:

1. Erstellen Sie eine VPC Sicherheitsgruppe für Ihren Cluster. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf die Cluster-Instances zu beschränken. Sie können beispielsweise eine benutzerdefinierte Regel für diese Sicherheitsgruppe erstellen, die den TCP Zugriff über den Port, den Sie dem Cluster bei der Erstellung zugewiesen haben, und über eine IP-Adresse, die Sie für den Zugriff auf den Cluster verwenden, ermöglicht.

Der Standardport für Valkey- oder OSS Redis-Cluster und Replikationsgruppen ist. 6379

Important

ElastiCache Amazon-Sicherheitsgruppen gelten nur für Cluster, die nicht in einer Amazon Virtual Private Cloud-Umgebung laufen (VPC). Wenn Sie ein Cluster in einer Amazon Virtual Private Cloud betreiben, ist die Option Sicherheitsgruppen im Navigationsbereich der Konsole nicht verfügbar.

Wenn Sie Ihre ElastiCache Knoten in einem Amazon betreibenVPC, kontrollieren Sie den Zugriff auf Ihre Cluster mit VPC Amazon-Sicherheitsgruppen, die sich von ElastiCache Sicherheitsgruppen unterscheiden. Weitere Informationen zur Verwendung

ElastiCache in einem Amazon VPC finden Sie unter [Amazon VPCs und ElastiCache Sicherheit](#)

- Erstellen Sie eine VPC Sicherheitsgruppe für Ihre EC2 Instances (Web- und Anwendungsserver). Diese Sicherheitsgruppe kann bei Bedarf den Zugriff auf die EC2 Instanz aus dem Internet über die Routingtabelle VPC von ermöglichen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, um den TCP Zugriff auf die EC2 Instance über Port 22 zu ermöglichen.
- Erstellen Sie in der Sicherheitsgruppe für Ihren Cluster benutzerdefinierte Regeln, die Verbindungen von der Sicherheitsgruppe aus zulassen, die Sie für Ihre EC2 Instances erstellt haben. Damit wird jedem Mitglied der Sicherheitsgruppe der Zugriff auf die DB-Instances gestattet.

Note

Wenn Sie planen, [Local Zones](#) zu nutzen, stellen Sie sicher, dass Sie sie aktiviert haben. Wenn Sie eine Subnetzgruppe in dieser lokalen Zone erstellen, VPC wird Ihre auf diese lokale Zone ausgedehnt und Sie VPC behandeln das Subnetz wie jedes Subnetz in einer anderen Availability Zone. Alle relevanten Gateways und Routing-Tabellen werden automatisch angepasst.

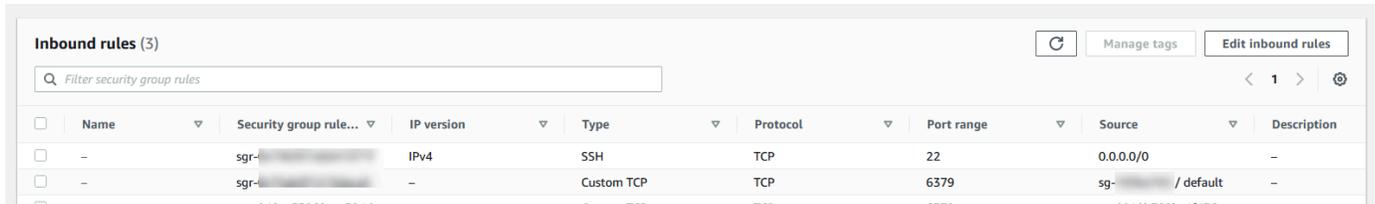
Um eine Regel in einer Sicherheitsgruppe zu erstellen, die Verbindungen von einer anderen VPC Sicherheitsgruppe aus zulässt

- Melden Sie sich bei der AWS Management Console an und öffnen Sie die VPC Amazon-Konsole unter <https://console.aws.amazon.com/vpc>.
- Wählen Sie im Navigationsbereich Security Groups (Sicherheitsgruppen) aus.
- Wählen Sie eine Sicherheitsgruppe aus oder erstellen Sie eine, die Sie für Ihre Cluster-Instances verwenden werden. Wählen Sie unter Inbound Rules (Eingangsregeln) die Option Edit Inbound Rules (Eingangsregeln bearbeiten) und dann Add Rule (Regeln hinzufügen). Diese Sicherheitsgruppe gewährt Mitgliedern einer anderen Sicherheitsgruppe Zugriff.
- Wählen Sie unter Typ die Option Benutzerdefinierte TCP Regel aus.
 - Geben Sie für Port Range den Port an, den Sie beim Erstellen des Clusters verwendet haben.

Der Standardport für Valkey- oder OSS Redis-Cluster und Replikationsgruppen ist. 6379

- b. Geben Sie in das Feld Source die ersten Zeichen der ID der Sicherheitsgruppe ein. Wählen Sie aus der Liste die Sicherheitsgruppe aus, die Sie für Ihre EC2 Amazon-Instances verwenden möchten.

5. Wählen Sie Save, wenn Sie fertig sind.



Wenn Sie den Zugriff aktiviert haben, können Sie eine Verbindung zum Knoten herstellen, wie im nächsten Abschnitt beschrieben.

Informationen zum Zugriff auf Ihren ElastiCache Cluster von einem anderen AmazonVPC, einer anderen AWS Region oder sogar von Ihrem Unternehmensnetzwerk aus finden Sie im Folgenden:

- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC](#)
- [Zugriff auf ElastiCache Ressourcen von außen AWS](#)

Schritt 4. Connect zum Knoten des Clusters her

Bevor Sie fortfahren, müssen Sie [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) abschließen.

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

Eine EC2 Amazon-Instance kann nur dann eine Verbindung zu einem Clusterknoten herstellen, wenn Sie sie dazu autorisiert haben.

Finden Sie Ihre Knotenendpunkte

Wenn sich Ihr Cluster im Status „Verfügbar“ befindet und Sie den Zugriff darauf autorisiert haben, können Sie sich bei einer EC2 Amazon-Instance anmelden und eine Verbindung zum Cluster herstellen. Hierzu müssen Sie zuerst den Endpunkt bestimmen.

Finden Sie die Endpunkte eines Valkey-Clusters (Cluster-Modus deaktiviert) (Konsole)

Wenn ein Valkey-Cluster (Clustermodus deaktiviert) nur einen Knoten hat, wird der Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge verwendet. Wenn ein Cluster mehrere Knoten enthält, gibt es drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte.

Der primäre Endpunkt ist ein DNS Name, der immer zum primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Leser-Endpunkt teilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem ElastiCache Cluster auf. Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Reader-Endpunkte sicherzustellen.

Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS Datensatz, der im Round-Robin-Verfahren in die IP-Adresse eines der Replikatknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

So finden Sie die Endpunkte eines Valkey-Clusters (Clustermodus deaktiviert)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Caches oder OSSRedis-Caches aus.

Der Cluster-Bildschirm wird mit einer Liste angezeigt, die alle vorhandenen OSS serverlosen Valkey- oder Redis-Caches, Valkey-Cluster (Cluster-Modus deaktiviert) und Valkey-Cluster (Cluster-Modus aktiviert) enthält. Wählen Sie den Cluster, den Sie im Abschnitt [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#) erstellt haben.

3. Um den primären und/oder den Reader-Endpunkt des Clusters zu ermitteln, wählen Sie den Namen des Clusters aus (nicht das Optionsfeld).

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 [Icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [Icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Primär- und Reader-Endpunkte für einen Valkey-Cluster (Cluster-Modus deaktiviert)

Wenn sich im Cluster nur ein Knoten befindet, ist kein primärer Endpunkt vorhanden. Sie können mit dem nächsten Schritt fortfahren.

4. Wenn der Valkey-Cluster (Cluster-Modus deaktiviert) über Replikatknoten verfügt, können Sie die Replikatknoten-Endpunkte des Clusters finden, indem Sie den Namen des Clusters und dann die Registerkarte Knoten auswählen.

Auf dem angezeigten Knotenbildschirm wird jeder Knoten im Cluster, –primärer Knoten und Replikate, – mit seinem Endpunkt aufgelistet.

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

Knotenendpunkte für einen Valkey-Cluster (Clustermodus deaktiviert)

5. So kopieren Sie einen Endpunkt in Ihre Zwischenablage:
 - a. Suchen Sie einen Endpunkt nach dem anderen nach dem zu kopierenden Endpunkt.
 - b. Wählen Sie das Kopiersymbol direkt vor dem Endpunkt aus.

Der Endpunkt wird nun in Ihre Zwischenablage kopiert. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Mit Knoten verbinden](#).

Ein primärer Valkey-Endpunkt (Clustermodus deaktiviert) sieht etwa wie folgt aus. Abhängig davon, ob die Verschlüsselung während der Übertragung aktiviert ist oder nicht, unterscheidet sich dies.

Verschlüsselung während der Übertragung nicht aktiviert

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

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

Verschlüsselung während der Übertragung aktiviert

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

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

Informationen zu den Endpunkten finden Sie im entsprechenden Thema für die Engine und den Cluster-Typ, die von Ihnen ausgeführt werden.

- [Verbindungsendpunkte finden in ElastiCache](#)
- [Finden von Endpunkten für einen Valkey- oder Redis-Cluster OSS \(Clustermodus aktiviert\) \(Konsole\)](#)—Sie benötigen den Konfigurationsendpunkt des Clusters.
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Endpunkte finden \(\) ElastiCache API](#)

Stellen Sie eine Connect zu einem Valkey- oder OSS Redis-Cluster oder einer Replikationsgruppe her (Linux)

Da Sie nun den benötigten Endpunkt haben, können Sie sich bei einer EC2 Instanz anmelden und eine Verbindung zum Cluster oder zur Replikationsgruppe herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Die neueste Version von `valkey-cli` unterstützt auch/für die Verbindung von Clustern mit aktivierter SSL Verschlüsselung/Authentifizierung/TLS.

Im folgenden Beispiel werden EC2 Amazon-Instances verwendet, auf denen Amazon Linux und Amazon Linux 2 ausgeführt werden. Einzelheiten zur Installation und Kompilierung von Valkey-Cli mit anderen Linux-Distributionen finden Sie in der Dokumentation für Ihr spezielles Betriebssystem.

Note

Dieser Prozess umfasst das Testen einer Verbindung mit dem Valkey-Cli-Hilfsprogramm nur für den ungeplanten Gebrauch. [Eine Liste der unterstützten Valkey- und OSS Redis-Clients finden Sie in der Valkey-Dokumentation.](#) Beispiele für die Verwendung von `with` finden Sie unter AWS SDKs. ElastiCache [Tutorials: Erste Schritte mit Python und ElastiCache](#)

Herstellen einer Verbindung zu einem unverschlüsselten Cluster mit deaktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen und zu ersetzen *primary-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Valkey oder Redis OSS ist 6379.)

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

Das Ergebnis einer Valkey- oder OSS Redis-Eingabeaufforderung sieht in etwa wie folgt aus:

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

2. Sie können jetzt Valkey- oder Redis-Befehle ausführen. OSS

```
set x Hello
OK

get x
"Hello"
```

Herstellen einer Verbindung mit einem unverschlüsselten Cluster mit aktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen und zu ersetzen *configuration-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Valkey oder Redis OSS ist 6379.)

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

Note

Im vorherigen Befehl aktiviert die Option `-c` den Clustermodus nach [- und -ASK](#) Umleitungen. `MOVED`

Das Ergebnis einer Valkey- oder OSS Redis-Eingabeaufforderung sieht in etwa wie folgt aus:

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

2. Sie können jetzt Valkey- oder Redis-Befehle ausführen. OSS Beachten Sie, dass die Umleitung erfolgt, weil Sie diese mit der Option `-c` aktiviert haben. Wenn die Umleitung nicht aktiviert ist, gibt der Befehl den Fehler zurück. `MOVED` Weitere Informationen zu dem `MOVED` Fehler finden Sie unter [OSSRedis-Clusterspezifikation](#).

```
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"
```

Herstellen einer Verbindung mit einem Cluster mit aktivierter Verschlüsselung/Authentifizierung

Standardmäßig verwendet valkey-cli eine unverschlüsselte TCP Verbindung, wenn eine Verbindung zu Valkey oder Redis hergestellt wird. OSS Die Option BUILD_TLS=yes aktiviert SSL/zum TLS Zeitpunkt der Valkey-CLI-Kompilierung, wie im vorherigen Abschnitt gezeigt. [Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein](#) Die Aktivierung AUTH ist optional. Sie müssen jedoch die Verschlüsselung bei der Übertragung aktivieren, um sie zu aktivieren AUTH. Weitere Informationen zur ElastiCache Verschlüsselung und Authentifizierung finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#).

Note

Sie können die Option `--tls` mit valkey-cli verwenden, um eine Verbindung zu verschlüsselten Clustern herzustellen, die im Clustermodus aktiviert oder deaktiviert sind. Wenn für einen Cluster ein AUTH Token festgelegt ist, können Sie die Option verwenden, `-a` um ein Passwort anzugeben. AUTH

Stellen Sie in den folgenden Beispielen sicher, dass Sie Folgendes ersetzen *cluster-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Valkey oder Redis OSS ist 6379.)

Verbindung zu verschlüsselten Clustern mit Cluster-Modus deaktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

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

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

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

Verbindung zu verschlüsselten Clustern mit Cluster-Modus aktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

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

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

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

Nachdem Sie eine Verbindung zum Cluster hergestellt haben, können Sie die OSS Befehle Valkey oder Redis ausführen, wie in den vorherigen Beispielen für unverschlüsselte Cluster gezeigt.

valkey-cli-Alternative

Wenn für den Cluster der Clustermodus nicht aktiviert ist und Sie für einen kurzen Test eine Verbindung zum Cluster herstellen müssen, ohne die Valkey-CLI-Kompilierung durchzuführen, können Sie Telnet oder OpenSSL verwenden. Achten Sie bei den folgenden Beispielbefehlen darauf, Folgendes zu ersetzen *cluster-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Valkey oder Redis OSS ist 6379.)

Im folgenden Beispiel wird eine Verbindung zu einem deaktivierten Cluster im Cluster-Modus mit Verschlüsselung und / oder Authentifizierung hergestellt:

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

Wenn für den Cluster ein Passwort festgelegt ist, stellen Sie zunächst eine Verbindung mit dem Cluster her. Nachdem Sie eine Verbindung hergestellt haben, authentifizieren Sie den Cluster mit

dem folgenden Befehl, und drücken Sie dann die Enter-Taste. Ersetzen Sie im folgenden Beispiel *your-password* mit dem Passwort für Ihren Cluster.

```
Auth your-password
```

Im folgenden Beispiel wird eine Verbindung zu einem deaktivierten Cluster im Cluster-Modus mit Verschlüsselung und / oder Authentifizierung hergestellt:

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

Stellen Sie eine Connect zu einem Valkey- oder OSS Redis-Cluster oder einer Replikationsgruppe her (Windows)

Um von einer Windows-Instanz aus mithilfe von Valkey oder Redis eine Verbindung zum Valkey- oder OSS Redis-Cluster herzustellen, müssen Sie das valkey-cli-Paket herunterladen und valkey-cli.exe verwenden OSSCLI, um von einer EC2 Windows-Instanz aus eine Verbindung zum Valkey - CLI oder Redis-Cluster herzustellen. OSS EC2

Im folgenden Beispiel verwenden Sie das Valkey-Cli-Hilfsprogramm, um eine Verbindung zu einem Cluster herzustellen, für den keine Verschlüsselung aktiviert ist und auf dem Valkey oder Redis ausgeführt werden. OSS Weitere Informationen zu Valkey oder Redis OSS und den verfügbaren Befehlen finden Sie unter Valkey- und Redis-Befehle auf der [Valkey-Website](#). OSS

So stellen Sie mithilfe von Valkey-CLI eine Verbindung zu einem Valkey- oder OSS Redis-Cluster her, der nicht verschlüsselungsfähig ist

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her. Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2Erste Schritte“](#).
2. Kopieren Sie den Link und fügen Sie ihn <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> in einen Internetbrowser ein, um die ZIP-Datei für den Valkey-Client aus der verfügbaren Version unter herunterzuladen GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Extrahieren Sie die ZIP-Datei in den gewünschten Ordner/Pfad.

Öffnen Sie die Befehlszeile, wechseln Sie in das Valkey-Verzeichnis und führen Sie den Befehl aus. `c:\Valkey>valkey-cli -h Redis_Cluster_Endpoint -p 6379`

Beispielsweise:

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

3. Führen Sie die Befehle Valkey oder OSS Redis aus.

Sie sind jetzt mit dem Cluster verbunden und können Valkey- oder OSS Redis-Befehle wie die folgenden ausführen.

```
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
```

Wie geht es weiter?

Nachdem Sie die Übung Erste Schritte ausprobiert haben, können Sie in den folgenden Abschnitten mehr über ElastiCache und verfügbare Tools erfahren:

- [Erste Schritte mit AWS](#)
- [Tools für Amazon Web Services](#)
- [AWS -Befehlszeilenschnittstelle](#)
- [ElastiCache API Amazon-Referenz](#)

Nachdem Sie die Übung Erste Schritte abgeschlossen haben, können Sie die folgenden Abschnitte lesen, um mehr über die ElastiCache Verwaltung zu erfahren:

- [Auswahl der Knotengröße](#)

Das Cache muss groß genug für alle Daten sein, die in das Cache aufgenommen werden sollen. Dabei möchten Sie auch nicht für mehr Cache bezahlen, als Sie benötigen. Verwenden Sie dieses Thema, um zu erfahren, wie Sie die beste Knotengröße auswählen.

- [ElastiCache bewährte Methoden und Caching-Strategien](#)

Identifizieren und Beheben von Problemen, die die Effizienz Ihres Clusters beeinträchtigen können.

Entwerfen Sie Ihren eigenen ElastiCache (Redis-OSS) Cluster

Im Folgenden sind die einmaligen Aktionen aufgeführt, die Sie ergreifen müssen, um Ihren eigenen ElastiCache (Redis-OSS) Cluster zu entwerfen.

Weitere Informationen zur Einrichtung finden ElastiCache Sie unter [einrichten ElastiCache](#).

Themen

- [Schritt 1: Erstellung einer Subnetzgruppe](#)
- [Schritt 2: Erstellen eines Clusters](#)
- [Schritt 3: Zugriff auf den Cluster autorisieren](#)
- [Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten](#)

Schritt 1: Erstellung einer Subnetzgruppe

Erstellen Sie eine Subnetz-Gruppe, bevor Sie einen Cluster erstellen. Eine Cache-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie möglicherweise für Ihre Cache-Cluster in einer angeben möchten. VPC Wenn Sie einen Cache-Cluster in einem startenVPC, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache Verwendet dann diese Cache-Subnetzgruppe, um jedem Cache-Knoten im Cluster IP-Adressen innerhalb dieses Subnetzes zuzuweisen.

Wenn Sie eine neue Subnetzgruppe erstellen, notieren Sie sich die Anzahl der verfügbaren IP-Adressen. Wenn das Subnetz nur über wenige freie IP-Adressen verfügt, beschränkt dies auch die Anzahl der neuen Knoten, die Sie zu dem Cluster hinzufügen können. Um dieses Problem zu lösen, können Sie einer Subnetzgruppe weitere Subnetze zuweisen, um ausreichend IP-Adressen in der Availability Zone Ihres Clusters bereitzustellen. Danach können Sie dem Cluster weitere Knoten hinzufügen.

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Subnetzgruppe mit dem Namen mysubnetgroup (Konsole) und die AWS CLI erstellen.

Erstellen einer Subnetzgruppe (Konsole)

Im folgenden Verfahren wird das Erstellen einer Subnetzgruppe (Konsole) erläutert.

Erstellen einer DB-Sicherheitsgruppe (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Klicken Sie in der Navigationsliste auf Subnet Groups.
3. Klicken Sie auf Create Subnet Group (Subnetzgruppe ändern).
4. Gehen Sie im Assistenten Create Subnet Group wie folgt vor. Wenn Sie alle gewünschten Einstellungen vorgenommen haben, klicken Sie auf Yes, Create.
 - a. Geben Sie im Feld Name einen Namen für Ihre Subnetzgruppe ein.
 - b. Geben Sie im Feld Description eine Beschreibung für Ihre Subnetzgruppe ein.
 - c. Wählen Sie im Feld VPCID den Amazon VPC aus, den Sie erstellt haben.
 - d. Wählen Sie unter Availability Zone und Subnetz-ID die Availability Zone oder [Local Zone](#) und die ID Ihres privaten Subnetzes aus und klicken Sie auf Hinzufügen.

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-██████████		172.31.32.0/20
us-east-1c	subnet-██████████		172.31.0.0/20
us-east-1d	subnet-██████████		172.31.80.0/20

5. Klicken Sie in der angezeigten Bestätigungsmeldung auf Close.

Ihre neue Subnetzgruppe wird in der Liste der Subnetzgruppen der Konsole angezeigt. ElastiCache Unten im Fenster können Sie die Subnetzgruppe auswählen, um Details wie die der Gruppe zugeordneten Subnetze anzuzeigen.

Erstellen einer Subnetzgruppe (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `create-cache-subnet-group` ein, um eine Subnetzgruppe zu erstellen.

Für Linux, macOS oder Unix:

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

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

Für Windows:

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

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

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

Weitere Informationen finden Sie im Thema AWS CLI [create-cache-subnet-group](#).

Schritt 2: Erstellen eines Clusters

Bevor Sie einen Cluster für die Produktion erstellen, müssen Sie natürlich überlegen, wie Sie den Cluster entsprechend Ihren geschäftlichen Anforderungen konfigurieren. Diese Probleme werden im [Einen Cluster vorbereiten in ElastiCache](#)-Abschnitt adressiert. Für die Zwecke dieser Übung „Erste Schritte“ erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus. Sie können die Standardkonfigurationswerte dort akzeptieren, wo sie zutreffen.

Der Cluster, den Sie gleich starten werden, wird live sein und nicht in einer Sandbox ausgeführt. Es fallen die ElastiCache Standardnutzungsgebühren für die Instance an, bis Sie sie löschen. Die Gesamtkosten werden minimal sein (meistens geringer als ein Dollar), wenn Sie diese Übung in einer

Sitzung durchlaufen und den Cluster löschen, sobald Sie die Übung abgeschlossen haben. Weitere Informationen zu den ElastiCache Nutzungsraten finden Sie auf [Amazon ElastiCache](#).

Ihr Cluster wird in einer virtuellen privaten Cloud (VPC) gestartet, die auf dem VPC Amazon-Service basiert.

Erstellen eines Redis-Clusters OSS (Cluster-Modus deaktiviert) (Konsole)

Um einen Redis-Cluster OSS (Clustermodus deaktiviert) mithilfe der Konsole zu erstellen ElastiCache

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der Sie diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Erstellen VPC und folgen Sie den unter [Erstellen einer virtuellen privaten Cloud \(\) VPC](#) beschriebenen Schritten.
5. Wählen Sie auf der ElastiCache Dashboard-Seite Valkey-Cache oder OSSRedis-Cache aus. In dieser Übung wählen wir OSSRedis-Cache und dann Redis-Cache erstellen. OSS
6. Führen Sie unter Cluster settings (Cluster-Einstellungen) die folgenden Schritte aus:
 - a. Wählen Sie Configure and create a new cluster (Neuen Cluster konfigurieren und erstellen) aus.
 - b. Wählen Sie für Cluster mode (Cluster-Modus) Disabled (Deaktiviert) aus.
 - c. Geben Sie für Cluster info (Cluster-Info) einen Wert für Name (Name) ein.
 - d. (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
7. Unter Location (Speicherort):

AWS Cloud

1. Wir empfehlen für AWS Cloud die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache \(Redis OSS\)](#) mit Multi-AZ.
2. Unter Cluster settings (Cluster-Einstellungen):
 - a. Wählen Sie unter Engine version (Engine-Version) eine verfügbare Version aus.

- b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.
- c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis-Parameter OSS](#) und [Eine ElastiCache Parametergruppe erstellen](#).

 Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parteigruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼)). Wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

- e. Wählen Sie unter Number of replicas (Anzahl der Repliken) die für diesen Cluster gewünschte Anzahl von Read Replicas aus. Wenn Multi-AZ aktiviert ist, muss die Zahl zwischen 1–5 liegen.
3. Unter Connectivity (Konnektivität)
 - a. Wählen Sie als Network type (Netzwerktyp) die IP-Version(en) aus, die dieser Cluster unterstützen soll.
 - b. Wählen Sie für Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten

zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only. IPv6

Geben Sie beim Erstellen einer neuen Subnetzgruppe die ID ein, zu der sie gehört. VPC

Weitere Informationen finden Sie unter:

- [Auswahl eines Netzwerktyps in ElastiCache.](#)
- [Erstellen Sie ein Subnetz in Ihrem. VPC](#)

Wenn Sie [Verwenden von lokalen Zonen mit ElastiCache](#) sind, müssen Sie ein Subnetz erstellen oder auswählen, das in der lokalen Zone liegt.

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen.](#)

4. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:

- Keine Präferenz — ElastiCache wählt die Availability Zone.
- Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache.](#)

5. Wählen Sie Next (Weiter).

6. Unter Erweiterte Redis-Einstellungen OSS

- Für Security (Sicherheit):
 - i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:
 - Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand.](#)

Note

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie vom Kunden verwalteter AWS KMS Schlüssel und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Vom Kunden verwaltete Schlüssel verwenden von AWS KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für die OSS Redis-Engine-Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
 - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
 - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Benutzergruppen mit der Konsole verwalten und CLI](#).
 - AUTHStandardbenutzer — Ein Authentifizierungsmechanismus für Valkey- und Redis-Server. OSS Weitere Informationen finden Sie unter [AUTH](#)
- AUTH— Ein Authentifizierungsmechanismus für den OSS Redis-Server. Weitere Informationen finden Sie unter [AUTH](#).

Note

Für OSS Redis-Versionen zwischen 3.2.6 und höher, mit Ausnahme von Version 3.2.10, ist OSS AUTH Redis die einzige Option.

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre VPC verwenden oder eine neue erstellen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Sie VPC](#) im VPCAmazon-Benutzerhandbuch.

7. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zur Sicherung und Wiederherstellung finden Sie unter [Snapshot und Wiederherstellung](#).

8. (Optional) Geben Sie ein Wartungsfenster an. Das Wartungsfenster ist der Zeitraum, der in der Regel eine Stunde lang ist, jede Woche, für den ElastiCache die Systemwartung Ihres Clusters geplant wird. Sie können ElastiCache den Tag und die Uhrzeit für Ihr Wartungsfenster wählen lassen (keine Präferenz), oder Sie können Tag, Uhrzeit und Dauer selbst wählen (Wartungsfenster angeben). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Zeiten sind UCT Zeiten.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

9. (Optional) Für Protokolle:
 - Wählen Sie unter Protokollformat entweder Text oder aus JSON.
 - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppenamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppenamen oder Ihren Firehose-Streamnamen aus.
10. Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Vorschlagworten Sie Ihre Ressourcen ElastiCache](#).
11. Wählen Sie Weiter.
12. Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Sobald Sie bereit sind, klicken Sie auf Create (Erstellen).

On premises

1. Für On premises (On-Premises) empfehlen wir Ihnen, Auto-failover (Automatisches Failover) aktiviert zu lassen. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache \(RedisOSS\)](#) mit Multi-AZ
2. Um die Erstellung des Clusters abzuschließen, befolgen Sie die Schritte unter [Verwenden von Outposts](#).

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Redis-Clusters OSS (Cluster-Modus deaktiviert) (AWS CLI)

Example

Der folgende CLI Code erstellt einen Redis-Cache-Cluster OSS (Clustermodus deaktiviert) ohne Replikate.

Für Linux, macOS oder 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
```

Für 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
```

Informationen zum Arbeiten mit aktiviertem Cluster-Modus finden Sie in den folgenden Themen:

- Wenn Sie die Konsole verwenden möchten, lesen Sie [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#).
- Informationen zur Verwendung von finden Sie unter AWS CLI. [Erstellen eines Valkey- oder Redis-Clusters OSS \(Clustermodus aktiviert\) \(\)AWS CLI](#)

Schritt 3: Zugriff auf den Cluster autorisieren

In diesem Abschnitt wird davon ausgegangen, dass Sie mit dem Starten und Herstellen einer Verbindung zu EC2 Amazon-Instances vertraut sind. Weitere Informationen finden Sie im [Amazon EC2 Getting Started Guide](#).

Alle ElastiCache Cluster sind so konzipiert, dass sie von einer EC2 Amazon-Instance aus aufgerufen werden können. Das gängigste Szenario ist der Zugriff auf einen ElastiCache Cluster von einer EC2 Amazon-Instance in derselben Amazon Virtual Private Cloud (AmazonVPC), was bei dieser Übung der Fall sein wird.

Standardmäßig ist der Netzwerkzugriff auf Ihren Cluster auf das Konto beschränkt, über das er erstellt wurde. Bevor Sie von einer EC2 Instance aus eine Verbindung zu einem Cluster herstellen können, müssen Sie die EC2 Instance für den Zugriff auf den Cluster autorisieren. Die erforderlichen Schritte hängen davon ab, ob Sie Ihren Cluster in EC2 - VPC oder EC2 -Classic gestartet haben.

Der häufigste Anwendungsfall ist, wenn eine auf einer EC2 Instanz bereitgestellte Anwendung eine Verbindung zu einem Cluster in derselben VPC Instanz herstellen muss. Der einfachste Weg, den Zugriff zwischen EC2 Instanzen und Clustern auf derselben Ebene zu verwalten, VPC besteht darin, wie folgt vorzugehen:

1. Erstellen Sie eine VPC Sicherheitsgruppe für Ihren Cluster. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf die Cluster-Instances zu beschränken. Sie können beispielsweise eine benutzerdefinierte Regel für diese Sicherheitsgruppe erstellen, die den TCP Zugriff über den Port, den Sie dem Cluster bei der Erstellung zugewiesen haben, und über eine IP-Adresse, die Sie für den Zugriff auf den Cluster verwenden, ermöglicht.

Der Standardport für OSS Redis-Cluster und Replikationsgruppen ist 6379.

Important

ElastiCache Amazon-Sicherheitsgruppen gelten nur für Cluster, die nicht in einer Amazon Virtual Private Cloud-Umgebung ausgeführt werden (VPC). Wenn Sie ein Cluster in einer Amazon Virtual Private Cloud betreiben, ist die Option Sicherheitsgruppen im Navigationsbereich der Konsole nicht verfügbar.

Wenn Sie Ihre ElastiCache Knoten in einem Amazon betreiben VPC, kontrollieren Sie den Zugriff auf Ihre Cluster mit VPC Amazon-Sicherheitsgruppen, die sich von ElastiCache Sicherheitsgruppen unterscheiden. Weitere Informationen zur Verwendung

ElastiCache in einem Amazon VPC finden Sie unter [Amazon VPCs und ElastiCache Sicherheit](#)

- Erstellen Sie eine VPC Sicherheitsgruppe für Ihre EC2 Instances (Web- und Anwendungsserver). Diese Sicherheitsgruppe kann bei Bedarf den Zugriff auf die EC2 Instanz aus dem Internet über die Routingtabelle VPC von ermöglichen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, um den TCP Zugriff auf die EC2 Instance über Port 22 zu ermöglichen.
- Erstellen Sie in der Sicherheitsgruppe für Ihren Cluster benutzerdefinierte Regeln, die Verbindungen von der Sicherheitsgruppe aus zulassen, die Sie für Ihre EC2 Instances erstellt haben. Damit wird jedem Mitglied der Sicherheitsgruppe der Zugriff auf die DB-Instances gestattet.

Note

Wenn Sie planen, sie zu verwenden [Verwenden von lokalen Zonen mit ElastiCache](#), stellen Sie sicher, dass Sie sie aktiviert haben. Wenn Sie eine Subnetzgruppe in dieser lokalen Zone erstellen, VPC wird Ihre auf diese lokale Zone ausgedehnt und Sie VPC behandeln das Subnetz wie jedes Subnetz in einer anderen Availability Zone. Alle relevanten Gateways und Routing-Tabellen werden automatisch angepasst.

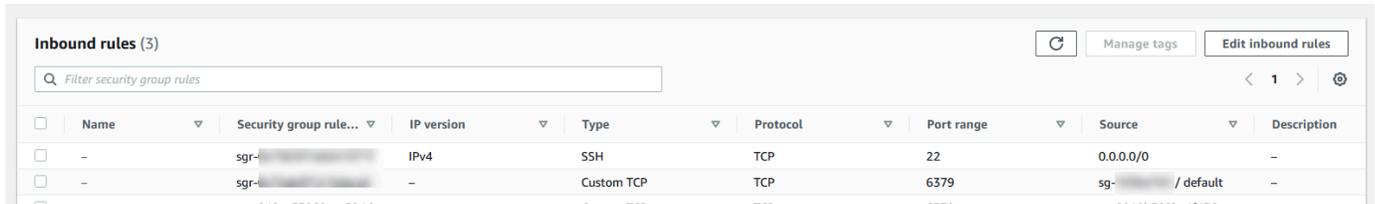
Um eine Regel in einer Sicherheitsgruppe zu erstellen, die Verbindungen von einer anderen VPC Sicherheitsgruppe aus zulässt

- Melden Sie sich bei der AWS Management Console an und öffnen Sie die VPC Amazon-Konsole unter <https://console.aws.amazon.com/vpc>.
- Wählen Sie im Navigationsbereich Security Groups (Sicherheitsgruppen) aus.
- Wählen Sie eine Sicherheitsgruppe aus oder erstellen Sie eine, die Sie für Ihre Cluster-Instances verwenden werden. Wählen Sie unter Inbound Rules (Eingangsregeln) die Option Edit Inbound Rules (Eingangsregeln bearbeiten) und dann Add Rule (Regeln hinzufügen). Diese Sicherheitsgruppe gewährt Mitgliedern einer anderen Sicherheitsgruppe Zugriff.
- Wählen Sie unter Typ die Option Benutzerdefinierte TCP Regel aus.
 - Geben Sie für Port Range den Port an, den Sie beim Erstellen des Clusters verwendet haben.

Der Standardport für OSS Redis-Cluster und Replikationsgruppen ist 6379.

- b. Geben Sie in das Feld Source die ersten Zeichen der ID der Sicherheitsgruppe ein. Wählen Sie aus der Liste die Sicherheitsgruppe aus, die Sie für Ihre EC2 Amazon-Instances verwenden möchten.

5. Wählen Sie Save, wenn Sie fertig sind.



Wenn Sie den Zugriff aktiviert haben, können Sie eine Verbindung zum Knoten herstellen, wie im nächsten Abschnitt beschrieben.

Informationen zum Zugriff auf Ihren ElastiCache Cluster von einem anderen AmazonVPC, einer anderen AWS Region oder sogar von Ihrem Unternehmensnetzwerk aus finden Sie im Folgenden:

- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC](#)
- [Zugriff auf ElastiCache Ressourcen von außen AWS](#)

Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten

Bevor Sie fortfahren, müssen Sie [Schritt 3: Zugriff auf den Cluster autorisieren](#) abschließen.

In diesem Abschnitt wird davon ausgegangen, dass Sie eine EC2 Amazon-Instance erstellt haben und eine Verbindung zu ihr herstellen können. Eine Anleitung dazu finden Sie im [Amazon EC2 Getting Started Guide](#).

Eine EC2 Amazon-Instance kann nur dann eine Verbindung zu einem Clusterknoten herstellen, wenn Sie sie dazu autorisiert haben.

Finden Sie Ihre Knotenendpunkte

Wenn sich Ihr Cluster im Status „Verfügbar“ befindet und Sie den Zugriff darauf autorisiert haben, können Sie sich bei einer EC2 Amazon-Instance anmelden und eine Verbindung zum Cluster herstellen. Hierzu müssen Sie zuerst den Endpunkt bestimmen.

Finden Sie die Endpunkte eines Valkey- oder Redis-Clusters OSS (Cluster-Modus deaktiviert) (Konsole)

Wenn ein Redis-Cluster OSS (Clustermodus deaktiviert) nur einen Knoten hat, wird der Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge verwendet. Wenn ein Cluster mehrere Knoten enthält, gibt es drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte.

Der primäre Endpunkt ist ein DNS Name, der immer zum primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Leser-Endpunkt teilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem ElastiCache (Redis OSS -) Cluster auf. Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache (Redis-OSS) Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Leser-Endpunkte sicherzustellen.

Note

Ein Leser-Endpoint ist kein Load Balancer. Es handelt sich um einen DNS Datensatz, der nach dem Round-Robin-Verfahren in die IP-Adresse eines der Replikatknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpoint werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

So finden Sie die Endpunkte eines Redis-Clusters OSS (Clustermodus deaktiviert)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich OSSRedis-Caches aus.

Der Cluster-Bildschirm wird mit einer Liste angezeigt, die alle vorhandenen Valkey- oder OSS Redis-Serverless-Caches, Redis-Cluster OSS (Cluster-Modus deaktiviert) und OSS Redis-Cluster (Cluster-Modus aktiviert) enthält. Wählen Sie den Cluster, den Sie im Abschnitt [Erstellen eines Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#) erstellt haben.

3. Um den primären und/oder den Reader-Endpoint des Clusters zu ermitteln, wählen Sie den Namen des Clusters aus (nicht das Optionsfeld).

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 [Icon] [redacted]-encrypted.llru6f.ng.0001.use1.cache.amazonaws.com:6379	Reader endpoint [Icon] [redacted]-encrypted-ro.llru6f.ng.0001.use1.cache.amazonaws.com:6379	ARN [redacted]	

Primär- und Reader-Endpunkte für einen Redis-Cluster OSS (Clustermodus deaktiviert)

Wenn sich im Cluster nur ein Knoten befindet, ist kein primärer Endpunkt vorhanden. Sie können mit dem nächsten Schritt fortfahren.

4. Wenn der Redis-Cluster OSS (Clustermodus deaktiviert) über Replikatknoten verfügt, können Sie die Replikatknoten-Endpunkte des Clusters finden, indem Sie den Namen des Clusters und dann die Registerkarte Knoten auswählen.

Auf dem angezeigten Knotenbildschirm wird jeder Knoten im Cluster, –primärer Knoten und Replikate, – mit seinem Endpunkt aufgelistet.

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

Knotenendpunkte für einen Redis-Cluster OSS (Clustermodus deaktiviert)

5. So kopieren Sie einen Endpunkt in Ihre Zwischenablage:
 - a. Suchen Sie einen Endpunkt nach dem anderen nach dem zu kopierenden Endpunkt.
 - b. Wählen Sie das Kopiersymbol direkt vor dem Endpunkt aus.

Der Endpunkt wird nun in Ihre Zwischenablage kopiert. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Mit Knoten verbinden](#).

Ein primärer Redis-Endpunkt OSS (Clustermodus deaktiviert) sieht etwa wie folgt aus. Abhängig davon, ob die Verschlüsselung während der Übertragung aktiviert ist oder nicht, unterscheidet sich dies.

Verschlüsselung während der Übertragung nicht aktiviert

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

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

Verschlüsselung während der Übertragung aktiviert

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

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

Informationen zu den Endpunkten finden Sie im entsprechenden Thema für die Engine und den Cluster-Typ, die von Ihnen ausgeführt werden.

- [Verbindungsendpunkte finden in ElastiCache](#)
- [Finden von Endpunkten für einen Valkey- oder Redis-Cluster OSS \(Clustermodus aktiviert\) \(Konsole\)](#)—Sie benötigen den Konfigurationsendpunkt des Clusters.
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Endpunkte finden \(\) ElastiCache API](#)

Stellen Sie eine Connect zu einem Valkey- oder OSS Redis-Cluster oder einer Replikationsgruppe her (Linux)

Da Sie nun den benötigten Endpunkt haben, können Sie sich bei einer EC2 Instanz anmelden und eine Verbindung zum Cluster oder zur Replikationsgruppe herstellen. Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen. Die neueste Version von `valkey-cli` unterstützt auch/für die Verbindung von Clustern mit aktivierter SSL Verschlüsselung/Authentifizierung/TLS.

Im folgenden Beispiel werden EC2 Amazon-Instances verwendet, auf denen Amazon Linux und Amazon Linux 2 ausgeführt werden. Einzelheiten zur Installation und Kompilierung von Valkey-Cli mit anderen Linux-Distributionen finden Sie in der Dokumentation für Ihr spezielles Betriebssystem.

Note

Dieser Prozess umfasst das Testen einer Verbindung mit dem Valkey-Cli-Hilfsprogramm nur für den ungeplanten Gebrauch. [Eine Liste der unterstützten Clients finden Sie in der Valkey-Dokumentation](#). Beispiele für die Verwendung von AWS SDKs with finden Sie ElastiCache unter [Tutorials: Erste Schritte mit Python und ElastiCache](#).

Herstellen einer Verbindung zu einem unverschlüsselten Cluster mit deaktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen und zu ersetzen *primary-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Valkey und Redis OSS ist 6379.)

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

Das Ergebnis in einer Befehlszeile sieht in etwa wie folgt aus:

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

2. Sie können jetzt die Befehle Valkey und Redis OSS ausführen.

```
set x Hello
OK

get x
"Hello"
```

Herstellen einer Verbindung mit einem unverschlüsselten Cluster mit aktiviertem Cluster-Modus

1. Führen Sie den folgenden Befehl aus, um eine Verbindung zum Cluster herzustellen und zu ersetzen *configuration-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Valkey und Redis OSS ist 6379.)

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

Note

Im vorherigen Befehl aktiviert die Option `-c` den Clustermodus nach [- und -ASK](#) Umleitungen. `MOVED`

Das Ergebnis in einer Befehlszeile sieht in etwa wie folgt aus:

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

2. Sie können jetzt die Befehle Valkey und Redis OSS ausführen. Beachten Sie, dass die Umleitung erfolgt, weil Sie diese mit der Option `-c` aktiviert haben. Wenn die Umleitung nicht aktiviert ist, gibt der Befehl den Fehler zurück. `MOVED` Weitere Informationen zu dem `MOVED` Fehler finden Sie unter [Clusterspezifikation](#).

```
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"
```

Herstellen einer Verbindung mit einem Cluster mit aktivierter Verschlüsselung/Authentifizierung

Standardmäßig verwendet valkey-cli eine unverschlüsselte TCP Verbindung, wenn eine Verbindung zu Valkey und Redis hergestellt wird. OSS Die Option BUILD_TLS=yes aktiviert SSL/zum TLS Zeitpunkt der Valkey-CLI-Kompilierung, wie im vorherigen Abschnitt gezeigt. [Laden Sie den Befehlszeilenzugriff herunter und richten Sie ihn ein](#) Die Aktivierung AUTH ist optional. Sie müssen jedoch die Verschlüsselung bei der Übertragung aktivieren, um sie zu aktivieren AUTH. Weitere Informationen zur ElastiCache Verschlüsselung und Authentifizierung finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#).

Note

Sie können die Option `--tls` mit valkey-cli verwenden, um eine Verbindung zu verschlüsselten Clustern herzustellen, die im Clustermodus aktiviert oder deaktiviert sind. Wenn für einen Cluster ein AUTH Token festgelegt ist, können Sie die Option verwenden, `-a` um ein Passwort anzugeben. AUTH

Stellen Sie in den folgenden Beispielen sicher, dass Sie Folgendes ersetzen *cluster-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Redis OSS ist 6379.)

Verbindung zu verschlüsselten Clustern mit Cluster-Modus deaktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

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

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

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

Verbindung zu verschlüsselten Clustern mit Cluster-Modus aktiviert

Das folgende Beispiel verbindet zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung:

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

Das folgende Beispiel verbindet zu einem Cluster nur mit aktivierter Verschlüsselung:

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

Nachdem Sie eine Verbindung zum Cluster hergestellt haben, können Sie die OSS Befehle Valkey oder Redis ausführen, wie in den vorherigen Beispielen für unverschlüsselte Cluster gezeigt.

Eine Alternative zu valkey-cli oder Redis-CLI

Wenn für den Cluster der Clustermodus nicht aktiviert ist und Sie für einen kurzen Test eine Verbindung zum Cluster herstellen müssen, ohne die Valkey-CLI- oder Redis-CLI-Kompilierung durchzuführen, können Sie Telnet oder OpenSSL verwenden. Achten Sie bei den folgenden Beispielbefehlen darauf, Folgendes zu ersetzen *cluster-endpoint* and *port number* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Redis OSS ist 6379.)

Im folgenden Beispiel wird eine Verbindung zu einem deaktivierten Cluster im Cluster-Modus mit Verschlüsselung und / oder Authentifizierung hergestellt:

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

Wenn für den Cluster ein Passwort festgelegt ist, stellen Sie zunächst eine Verbindung mit dem Cluster her. Nachdem Sie eine Verbindung hergestellt haben, authentifizieren Sie den Cluster mit

dem folgenden Befehl, und drücken Sie dann die Enter-Taste. Ersetzen Sie im folgenden Beispiel *your-password* mit dem Passwort für Ihren Cluster.

```
Auth your-password
```

Im folgenden Beispiel wird eine Verbindung zu einem deaktivierten Cluster im Cluster-Modus mit Verschlüsselung und / oder Authentifizierung hergestellt:

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

Stellen Sie eine Connect zu einem Valkey- oder OSS Redis-Cluster oder einer Replikationsgruppe her (Windows)

Um von einer EC2 Windows-Instanz aus mithilfe von Valkey oder Redis eine Verbindung zum Cluster herzustellen, müssen Sie das Paket valkey-cli herunterladen und valkey-cli.exe verwenden OSSCLI, um von einer Windows-Instanz aus eine Verbindung zum Valkey - oder Redis-Cluster herzustellen.
OSS EC2

Im folgenden Beispiel verwenden Sie das Valkey-Cli-Hilfsprogramm, um eine Verbindung zu einem Cluster herzustellen, der nicht verschlüsselungsfähig ist und auf dem Valkey oder Redis ausgeführt werden. OSS Weitere Informationen zu Valkey und verfügbaren Befehlen finden Sie unter Valkey-Befehle auf der [Valkey-Website](#).

So stellen Sie mithilfe von Valkey-CLI eine Verbindung zu einem Valkey- oder OSS Redis-Cluster her, der nicht verschlüsselungsfähig ist

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her. Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2Erste Schritte“](#).
2. Kopieren Sie den Link und fügen Sie ihn <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> in einen Internetbrowser ein, um die ZIP-Datei für den OSS Redis-Client aus der verfügbaren Version unter herunterzuladen GitHub <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504>

Extrahieren Sie die ZIP-Datei in den gewünschten Ordner/Pfad.

Öffnen Sie die Befehlszeile, wechseln Sie in das Valkey-Verzeichnis und führen Sie den Befehl aus. `c:\Valkey>valkey-cli -h Valkey_Cluster_Endpoint -p 6379`

Beispielsweise:

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

3. Führen Sie die Befehle Valkey oder OSS Redis aus.

Sie sind jetzt mit dem Cluster verbunden und können Valkey- oder OSS Redis-Befehle wie die folgenden ausführen.

```
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
```

Löschen eines Clusters

Solange sich ein Cluster im Zustand `available` befindet, werden Ihnen dafür Gebühren berechnet. Dabei spielt es keine Rolle, ob Sie ihn aktiv nutzen oder ob nicht. Löschen Sie den Cluster, damit keine Gebühren mehr anfallen.

Warning

- Wenn Sie einen ElastiCache Cluster löschen, werden Ihre manuellen Snapshots beibehalten. Sie können auch einen letzten Snapshot erstellen, bevor der Cluster gelöscht wird. Automatisch erstellte Cache-Snapshots werden nicht aufbewahrt. Weitere Informationen finden Sie unter [Snapshot und Wiederherstellung](#).

- `CreateSnapshot` Zum Erstellen eines endgültigen Snapshots ist eine Genehmigung erforderlich. Ohne diese Genehmigung schlägt der API Aufruf mit einer `Access Denied` Ausnahme fehl.

Mit dem AWS Management Console

Mit dem folgenden Verfahren wird ein einzelner Cluster aus Ihrer Bereitstellung gelöscht. Um mehrere Cache-Cluster zu löschen, wiederholen Sie das Verfahren für jeden Cluster, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Cluster fertig gelöscht ist, bevor Sie den Vorgang zum Löschen eines anderen Clusters starten.

Löschen eines Clusters

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Engine-Dashboard Valkey oder OSS Redis aus.

Eine Liste aller Caches, die auf dieser Engine laufen, wird angezeigt.

3. Wählen Sie aus der Liste den Namen des zu löschenden Clusters aus. In diesem Fall der Name des von Ihnen unter [Schritt 2: Erstellen eines Clusters](#) erstellten Clusters.

Important

Sie können jeweils nur einen Cluster von der ElastiCache Konsole löschen. Werden mehrere Cluster ausgewählt, wird die Löschoperation deaktiviert.

4. Klicken Sie bei Actions auf Delete.
5. Geben Sie im Bestätigungsbildschirm Cluster löschen den Namen des Clusters ein und wählen Sie Endgültiges Backup aus. Wählen Sie dann Löschen, um den Cluster zu löschen, oder Abbrechen aus, wenn Sie den Cluster beibehalten möchten.

Wenn Sie Delete auswählen, ändert sich der Status des Clusters zu deleting.

Sobald dieser Cluster nicht mehr in der Cluster-Liste erscheint, fallen dafür keine Gebühren mehr an.

Mit dem AWS CLI

Der folgende Code löscht den Cache-Cluster `my-cluster`. In diesem Fall ersetzen Sie `my-cluster` mit dem Namen des Clusters, den Sie unter [Schritt 2: Erstellen eines Clusters](#) erstellt haben.

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

Die `delete-cache-cluster` CLI Aktion löscht nur einen Cache-Cluster. Um mehrere Cache-Cluster zu löschen, rufen Sie `delete-cache-cluster` für jeden Cache-Cluster auf, den Sie löschen möchten. Wenn Sie Ihren Cluster in einer VPC starten, die auf dem `-Service` basiert, können Sie außerhalb von darauf zugreifen.

Für Linux, macOS oder Unix:

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

Für Windows:

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

Weitere Informationen finden Sie im AWS CLI ElastiCache For-Thema [delete-cache-cluster](#).

Weitere ElastiCache Tutorials und Videos

Die folgenden Tutorials befassen sich mit Aufgaben, die für den ElastiCache Amazon-Benutzer von Interesse sind.

- [ElastiCache Videos](#)
- [Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einem Amazon VPC](#)

ElastiCache Videos

Hier finden Sie Videos, die Ihnen helfen, grundlegende und fortgeschrittene ElastiCache Amazon-Konzepte zu erlernen. Informationen zu AWS Schulungen finden Sie unter [AWS Schulung und Zertifizierung](#).

Themen

- [Einführungsvideos](#)
- [Weiterführende Videos](#)

Einführungsvideos

Die folgenden Videos stellen Ihnen Amazon vor ElastiCache.

Themen

- [AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache](#)
- [AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache](#)
- [AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache](#)
- [DAT204 — Entwicklung skalierbarer Anwendungen AWS ohne SQL Dienste \(re:Invent 2015\)](#)
- [DAT207 — Schnellere Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)AWS](#)

AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache

[AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache](#)

AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache

[AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache](#)

AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache

[AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache](#)

DAT204 — Entwicklung skalierbarer Anwendungen AWS ohne SQL Dienste (re:Invent 2015)

In dieser Sitzung besprechen wir die Vorteile von No SQL Databases und geben einen Überblick über die wichtigsten SQL No-Services, die von AWS Amazon DynamoDB und Amazon angeboten werden.

ElastiCache Anschließend hören wir von zwei führenden Kunden, Expedia und Mapbox, über ihre Anwendungsfälle und architektonischen Herausforderungen und darüber, wie sie diese mithilfe von AWS SQL No-Services bewältigt haben, einschließlich Entwurfsmustern und Best Practices. Nach Abschluss dieser Sitzung sollten Sie No SQL und seine leistungsstarken Funktionen besser verstehen und bereit sein, Ihre Datenbankprobleme mit Zuversicht anzugehen.

[DAT204 — Entwicklung skalierbarer Anwendungen AWS ohne SQL Dienste \(re:Invent 2015\)](#)

[DAT207 — Schnellere Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)AWS](#)

In diesem Video erfahren Sie, wie Sie Amazon verwenden können, ElastiCache um auf einfache Weise ein In-Memory-Caching-System bereitzustellen, um die Leistung Ihrer Anwendung zu beschleunigen. Wir zeigen Ihnen, wie Sie Amazon verwenden können ElastiCache , um die Latenz Ihrer Anwendungen zu verbessern und die Belastung Ihrer Datenbankserver zu reduzieren. Wir zeigen Ihnen auch, wie Sie eine Caching-Schicht aufbauen, die einfach zu verwalten und zu skalieren ist, wenn Ihre Anwendung wächst. In dieser Sitzung gehen wir auf verschiedene Szenarien und Anwendungsfälle ein, die von der Aktivierung von Caching profitieren können, und besprechen die von Amazon ElastiCache bereitgestellten Funktionen.

[DAT207 — Schnellere Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)](#)

Weiterführende Videos

Die folgenden Videos behandeln fortgeschrittenere ElastiCache Amazon-Themen.

Themen

- [Erfolgreiches Design mit den ElastiCache Best Practices von Amazon \(re:Invent 2020\)](#)
- [Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache \(re:Invent 2019\)](#)
- [Bewährte Methoden: Migration von OSS Redis-Clustern von Amazon EC2 zu ElastiCache \(re:Invent 2019\)](#)
- [Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Zuverlässiges und skalierbares Redis OSS in der Cloud mit Amazon ElastiCache \(re:Invent 2018\)](#)
- [ElastiCache Deep Dive: Entwurfsmuster für In-Memory-Datenspeicher \(re:Invent 2018\)](#)
- [DAT305 — Amazon ElastiCache Deep Dive \(re:Invent 2017\)](#)
- [DAT306 — Amazon ElastiCache Deep Dive \(re:Invent 2016\)](#)

- [DAT317—Wie IFTTT verwendet man ElastiCache \(RedisOSS\) zur Vorhersage von Ereignissen \(re:Invent 2016\)](#)
- [DAT407 — Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)
- [SDD402 — Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)
- [DAT307 — Tiefer Einblick in die ElastiCache Architektur und Entwurfsmuster von Amazon \(re:Invent 2013\)](#)

Erfolgreiches Design mit den ElastiCache Best Practices von Amazon (re:Invent 2020)

Angesichts des explosionsartigen Wachstums geschäftskritischer Echtzeitanwendungen, die auf Redis basieren, sind VerfügbarkeitOSS, Skalierbarkeit und Sicherheit zu den wichtigsten Überlegungen geworden. Lernen Sie bewährte Methoden kennen, um Amazon ElastiCache mit Online-Skalierung, Hochverfügbarkeit in Multi-AZ-Bereitstellungen und Sicherheitskonfigurationen auf Erfolgskurs zu bringen.

[Erfolgreiches Design mit den ElastiCache Best Practices von Amazon \(re:Invent 2020\)](#)

Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache (re:Invent 2019)

Angesichts der rasanten Zunahme der Cloud-Nutzung und der damit verbundenen neuen Szenarien benötigen Anwendungen Latenzzeiten im Mikrosekundenbereich und einen hohen Durchsatz, um Millionen von Anfragen pro Sekunde zu unterstützen. Für die Verwaltung von Daten für Echtzeitanwendungen haben sich die Entwickler bisher auf spezielle Hardware und Workarounds verlassen, z. B. auf plattenbasierte Datenbanken in Kombination mit Datenreduzierungstechniken. Diese Vorgehensweisen können kostspielig und nicht erweiterungsfähig sein. Erfahren Sie, wie Sie die Leistung von Echtzeitanwendungen steigern können, indem Sie das vollständig verwaltete In-Memory-Amazon ElastiCache für extreme Leistung, hohe Skalierbarkeit, Verfügbarkeit und Sicherheit nutzen.

[Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache \(re:Invent 2019\)](#)

Bewährte Methoden: Migration von OSS Redis-Clustern von Amazon EC2 zu ElastiCache (re:Invent 2019)

Es kann schwierig sein, OSS Redis-Cluster selbst zu verwalten. Sie müssen Hardware bereitstellen, Software patchen, Daten sichern und Workloads kontinuierlich überwachen. Mit der neu veröffentlichten Online-Migrationsfunktion für Amazon ElastiCache können Sie Ihre Daten jetzt ganz einfach von selbst gehostetem Redis OSS auf Amazon auf vollständig verwaltetes Amazon EC2 verschieben ElastiCache, wobei der Clustermodus deaktiviert ist. In dieser Sitzung erfahren Sie mehr

über das neue Online-Migrationstool, sehen sich eine Demo an und, was noch wichtiger ist, lernen Sie praktische Best Practices für eine reibungslose Migration zu Amazon ElastiCache kennen.

[Bewährte Methoden: Migration von OSS Redis-Clustern von Amazon EC2 zu ElastiCache \(re:Invent 2019\)](#)

Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 (re:Invent 2018)

Dream11 ist Indiens führendes Sport-Tech-Startup. Es hat eine wachsende Basis von mehr als 40 Millionen Nutzern, die mehrere Sportarten spielen, darunter Fantasy Cricket, Fußball und Basketball, und bedient derzeit eine Million gleichzeitige Nutzer, die drei Millionen Anfragen pro Minute mit einer Reaktionszeit von 50 Millisekunden stellen. In diesem Vortrag erklärt Dream11 CTO Amit Sharma, wie das Unternehmen Amazon Aurora und Amazon für den Umgang mit Flash-Verkehr verwendet, der ElastiCache sich innerhalb eines Antwortfensters von 30 Sekunden verdreifachen kann. Sharma spricht auch über die Skalierung von Transaktionen ohne Sperren und erklärt, wie man den Flash-Verkehr bewältigt und dabei fünf Millionen täglich aktive Nutzer bedient. Vollständiger Titel: AWS re:Invent 2018: Skalierung einer Fantasy-Sportplattform mit Amazon ElastiCache und Amazon Aurora () STP11

[Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 \(re:Invent 2018\)](#)

Zuverlässiges und skalierbares Redis OSS in der Cloud mit Amazon ElastiCache (re:Invent 2018)

In dieser Sitzung werden die Funktionen und Verbesserungen unseres OSS Redis-kompatiblen Dienstes Amazon ElastiCache (OSSRedis) behandelt. Wir behandeln wichtige Funktionen wie Redis OSS 5, Skalierbarkeits- und Leistungsverbesserungen, Sicherheit und Compliance und vieles mehr. Wir besprechen auch zukünftige Funktionen und Kundenfallstudien.

[Zuverlässiges und skalierbares Redis OSS in der Cloud mit Amazon ElastiCache \(re:Invent 2018\)](#)

ElastiCache Deep Dive: Entwurfsmuster für In-Memory-Datenspeicher (re:Invent 2018)

In dieser Sitzung bieten wir einen Blick hinter die Kulissen, um mehr über das Design und die Architektur von Amazon zu erfahren ElastiCache. Sehen Sie sich die gängigen Entwurfsmuster unserer Redis OSS - und Memcached-Angebote an und erfahren Sie, wie Kunden sie für die In-Memory-Datenverarbeitung verwenden, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. Wir überprüfen ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns.

[ElastiCache Deep Dive: Entwurfsmuster für speicherinterne Datenspeicher \(re:Invent 2018\)](#)

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

Schauen Sie hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich gängige Entwurfsmuster unserer Memcached- und OSS Redis-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen verwendet haben, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. In diesem Video gehen wir auf ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns ein.

Das Video zeigt Folgendes:

- ElastiCache (RedisOSS) Online-Resharding
- ElastiCache Sicherheit und Verschlüsselung
- ElastiCache (RedisOSS) Version 3.2.10

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

DAT306 — Amazon ElastiCache Deep Dive (re:Invent 2016)

Schauen Sie hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich gängige Entwurfsmuster unserer Memcached- und OSS Redis-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen verwendet haben, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. In dieser Sitzung besprechen wir ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns.

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

DAT317—Wie IFTTT verwendet man ElastiCache (RedisOSS) zur Vorhersage von Ereignissen (re:Invent 2016)

IFTTT ist ein kostenloser Service, der es Menschen ermöglicht, die Dienste, die sie lieben, besser zu nutzen, von der Automatisierung einfacher Aufgaben bis hin zur Veränderung der Art und Weise, wie jemand mit ihrem Zuhause interagiert und es steuert. IFTTT verwendet ElastiCache (RedisOSS), um den Verlauf von Transaktionsläufen und Zeitplanprognosen sowie Indizes für Protokoll dokumente auf Amazon S3 zu speichern. Sehen Sie sich diese Sitzung an, um zu erfahren, wie die Skripting-Leistung von Lua und die Datentypen von Redis OSS es Menschen ermöglichen, etwas zu erreichen, das sie anderswo nicht hätten erreichen können.

[DAT317—Wie IFTTT verwendet man ElastiCache \(RedisOSS\) zur Vorhersage von Ereignissen \(re:Invent 2016\)](#)

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

Werfen Sie einen Blick hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich die gängigen Entwurfsmuster unserer Memcached- und OSS Redis-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen eingesetzt und eine verbesserte Latenz und einen verbesserten Durchsatz für Anwendungen erzielt haben. In dieser Sitzung besprechen wir bewährte Verfahren, Entwurfsmuster und Anti-Pattern-Methoden im Zusammenhang mit Amazon ElastiCache.

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

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

In diesem Video untersuchen wir gängige Caching-Anwendungsfälle, die Memcached- und OSS Redis-Engines, Muster, anhand derer Sie ermitteln können, welche Engine für Ihre Anforderungen besser geeignet ist, konsistentes Hashing und vieles mehr, um schnelle, skalierbare Anwendungen zu entwickeln. Frank Wiebe, Principal Scientist bei Adobe, beschreibt, wie Adobe Amazon nutzt, ElastiCache um das Kundenerlebnis zu verbessern und sein Geschäft zu skalieren.

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

[DAT307 — Tiefer Einblick in die ElastiCache Architektur und Entwurfsmuster von Amazon \(re:Invent 2013\)](#)

In diesem Video untersuchen wir Caching, Caching-Strategien, Skalierung und Überwachung. Wir vergleichen auch die Memcached- und Redis-Engines. OSS Während dieser Sitzung besprechen wir auch Best Practices und Designmuster im Zusammenhang mit Amazon ElastiCache.

[DAT307 — Tiefer Einblick in die ElastiCache Architektur und Designmuster von Amazon \(AWS re:Invent 2013\).](#)

Knoten verwalten in ElastiCache

Ein Knoten ist der kleinste Baustein einer ElastiCache Amazon-Bereitstellung. Es handelt sich um einen sicheren, an das Netzwerk angeschlossenen Block mit fester Größe. RAM Auf jedem Knoten wird die Engine ausgeführt, je nachdem, welche Option beim Erstellen oder der letzten Änderung des Clusters ausgewählt wurde. Jeder Knoten hat seinen eigenen Namen und Port für den Domain Name Service (DNS). Es werden mehrere ElastiCache Knotentypen unterstützt, die jeweils unterschiedliche Mengen an zugeordnetem Speicher und Rechenleistung aufweisen.

Eine ausführliche Erörterung der zu verwendenden Knotengröße finden Sie unter [Auswahl der Knotengröße](#).

Im Allgemeinen verfügen Valkey- oder Redis-Bereitstellungen OSS (Clustermodus aktiviert) aufgrund der Unterstützung von Sharding über eine Reihe kleinerer Knoten. Im Gegensatz dazu haben Valkey- oder Redis-Bereitstellungen OSS (Clustermodus deaktiviert) weniger, dafür aber größere Knoten in einem Cluster. Eine ausführliche Erörterung der zu verwendenden Knotengröße finden Sie unter [Auswahl der Knotengröße](#).

Themen

- [ElastiCache Knotenstatus anzeigen](#)
- [Valkey- oder OSS Redis-Knoten und -Shards](#)
- [Mit Knoten verbinden](#)
- [Unterstützte Knotentypen](#)
- [Knoten neu starten](#)
- [Knoten ersetzen \(Valkey und OSS Redis\)](#)
- [Knoten ersetzen \(Memcached\)](#)
- [Reservierte Knoten](#)
- [Migration von Knoten der vorherigen Generation](#)

Einige wichtige Operationen, die Knoten umfassen, sind folgende:

- [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)
- [Knoten aus einem ElastiCache Cluster entfernen](#)
- [Skalierung ElastiCache](#)
- [Verbindungsendpunkte finden in ElastiCache](#)
- [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#)

ElastiCache Knotenstatus anzeigen

Über die [ElastiCache Konsole](#) können Sie schnell auf den Status Ihres ElastiCache Knotens zugreifen. Der Status eines ElastiCache Knotens gibt den Zustand des Knotens an. Sie können die folgenden Verfahren verwenden, um den ElastiCache Knotenstatus in der ElastiCache Amazon-Konsole, im AWS CLI Befehl oder im API Vorgang anzuzeigen.

Die möglichen Statuswerte für ElastiCache Knoten sind in der folgenden Tabelle aufgeführt. In dieser Tabelle ist auch angegeben, ob Ihnen der ElastiCache Knoten in Rechnung gestellt wird.

Typ	Berechnet	Beschreibung
available	Berechnet	Der ElastiCache Knoten ist fehlerfrei und verfügbar.
creating	Nicht berechnet	Der ElastiCache Knoten wird gerade erstellt. Während der Knoten erstellt wird, kann nicht auf ihn zugegriffen werden.
deleting	Nicht berechnet	Der ElastiCache Knoten wird gelöscht.
modifying	Berechnet	Der ElastiCache Knoten wird aufgrund einer Kundenanfrage zur Änderung des Knotens geändert.
updating	Berechnet	<p>Ein Aktualisierungsstatus gibt an, dass eine oder mehrere der folgenden Bedingungen für den ElastiCache Amazon-Knoten zutreffen:</p> <ul style="list-style-type: none"> • Der ElastiCache Knoten wird im Rahmen des Service-Updates gepatcht. Weitere Informationen zu den Service-Updates finden Sie auf der Hilfeseite Amazon ElastiCache Managed Maintenance and Service Updates. • Die VPC Sicherheitsgruppen werden für den

Typ	Berechnet	Beschreibung
		<p>ElastiCache Cluster aktualisiert.</p> <ul style="list-style-type: none">• Der ElastiCache Cluster wird hoch- oder herunterskaliert.• Die Konfigurationen für die Protokollzustellung werden für den ElastiCache Cluster geändert.• Ein Löschvorgang für den ElastiCache Knoten steht noch aus.• Das ElastiCache with Valkey- oder OSS Redis-Passwort wird aktualisiert/rotiert mit AWS Secrets Manager
rebooting cache cluster nodes	Berechnet	Der ElastiCache Knoten wird aufgrund einer Kundenanfrage oder eines ElastiCache Amazon-Prozesses, der einen Neustart des Knotens erfordert, neu gestartet.

Typ	Berechnet	Beschreibung
<code>incompatible_parameters</code>	Nicht berechnet	<p>Amazon ElastiCache kann den Knoten nicht starten, da die in der Parametergruppe des Knotens angegebenen Parameter nicht mit dem Knoten kompatibel sind. Machen Sie entweder die letzten Änderungen an den Parametern rückgängig oder stellen Sie ihre Kompatibilität mit dem Knoten her, um wieder auf den Knoten zugreifen zu können. Weitere Informationen zu den inkompatiblen Parametern finden Sie in der Ereignisliste für den ElastiCache Knoten.</p>
<code>incompatible_network</code>	Nicht berechnet	<p>Ein inkompatibler Netzwerkstatus bedeutet, dass eine oder mehrere der folgenden Bedingungen auf den Amazon-Knoten zutreffen: ElastiCache</p> <ul style="list-style-type: none">• In dem Subnetz, in dem der ElastiCache Knoten gestartet wurde, sind keine IP-Adressen verfügbar.• Das in der Subnetzgruppe erwähnte ElastiCache Subnetz ist in der Amazon Virtual Private Cloud (AmazonVPC) nicht mehr vorhanden.

Typ	Berechnet	Beschreibung
restore_failed	Nicht berechnet	<p>Ein Status „Wiederherstellung fehlgeschlagen“ bedeutet, dass einer der folgenden Punkte auf den Amazon-Knoten zutrifft: ElastiCache</p> <ul style="list-style-type: none">• Der Austausch des Knotens schlug wiederholt fehl, da die Instance-Kapazität nicht ausreichte. Dies ist in der Regel der Fall, wenn Knoten der vorherigen Generation ausgeführt werden. end-of-life Dies kann jedoch auch passieren, wenn Knoten der aktuellen Generation ersetzt werden, wenn AWS nicht genügend On-Demand-Kapazität zur Erfüllung Ihrer Anfrage in der angegebenen Availability Zone zur Verfügung steht. Weitere Informationen zum Reparieren oder Entfernen dieser Knoten finden Sie unter Migration von Knoten der vorherigen Generation.• Der angegebene RDB Snapshot konnte nicht wiederhergestellt werden.• Das AWS Konto für den ElastiCache Cluster wurde gesperrt.

Typ	Berechnet	Beschreibung
		<ul style="list-style-type: none">• Der Knoten ist ausgefallen und konnte nicht wiederhergestellt werden.
snapshotting	Berechnet	ElastiCache erstellt einen Snapshot des Valkey- oder OSS Redis-Knotens.

Den ElastiCache Knotenstatus mit der Konsole anzeigen

So zeigen Sie den Status eines ElastiCache Nodes mit der Konsole an:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis OSS Clusters oder Memcached Clusters aus. Die Seite Caches mit der Liste der Knoten wird angezeigt. ElastiCache Für die einzelnen Knoten wird der Statuswert angezeigt.
3. Sie können dann zur Registerkarte Service Updates für den Cache navigieren, um die Liste der für den Cache geltenden Service-Updates anzuzeigen.

Den ElastiCache Knotenstatus anzeigen mit dem AWS CLI

Um den ElastiCache Knoten und seine Statusinformationen mithilfe von anzuzeigen AWS CLI, verwenden Sie den `describe-cache-cluster` Befehl. Der folgende AWS CLI Befehl zeigt beispielsweise jeden ElastiCache Knoten an.

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

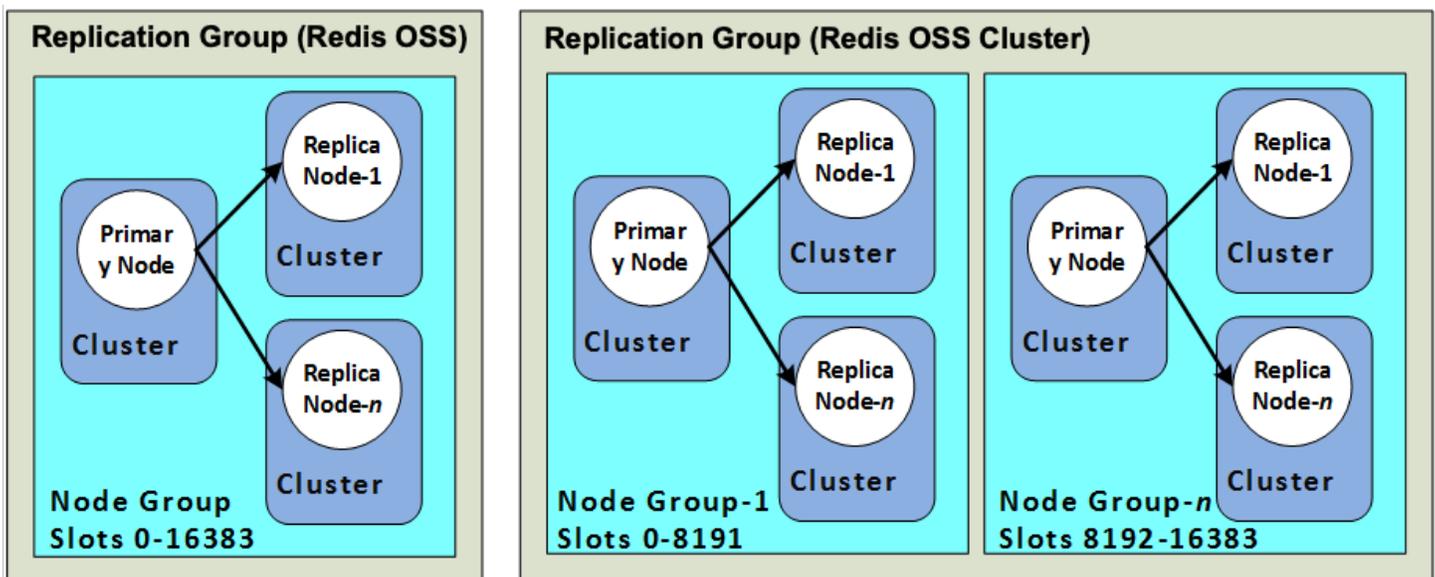
Den ElastiCache Knotenstatus anzeigen über den API

Um den Status des ElastiCache Nodes zu sehen, der Amazon verwendet ElastiCache API, rufen Sie die `DescribeCacheClusterOperation` mit dem `ShowCacheNodeInfo` Flag auf, um Informationen über die einzelnen Cache-Knoten abzurufen.

Valkey- oder OSS Redis-Knoten und -Shards

Ein Shard (in der API und CLI, einer Knotengruppe) ist eine hierarchische Anordnung von Knoten, die jeweils in einem Cluster zusammengefasst sind. Shards unterstützen die Replikation. Innerhalb eines Shards dient ein Knoten als Primärknoten mit Lese-/Schreibzugriff. Alle anderen Knoten in einem Shard dienen als schreibgeschützte Replikate des Primärknotens. Valkey oder Redis OSS Version 3.2 und höher unterstützen mehrere Shards innerhalb eines Clusters (in der API und CLI, einer Replikationsgruppe). Diese Unterstützung ermöglicht die Partitionierung Ihrer Daten in einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert).

Das folgende Diagramm veranschaulicht die Unterschiede zwischen einem Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) und einem Valkey- oder OSS Redis-Cluster (Cluster-Modus aktiviert).



Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) unterstützen die Replikation über Shards. Die API Operation [DescribeReplicationGroups](#) (CLI: [describe-replication-groups](#)) listet die Knotengruppen mit den Mitglieds-knoten, die Rolle des Knotens innerhalb der Knotengruppe und weitere Informationen auf.

Wenn Sie einen Valkey- oder OSS Redis-Cluster erstellen, geben Sie an, ob Sie einen Cluster mit aktiviertem Clustering erstellen möchten. Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) haben nie mehr als einen Shard. Dieser kann horizontal skaliert werden, indem Read Replica-Knoten hinzugefügt (bis zu insgesamt fünf) oder gelöscht werden. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#), [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#) oder [Löschen einer Read Replica für Valkey oder Redis OSS](#)

[\(Cluster-Modus deaktiviert\)](#). Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) können auch vertikal skaliert werden, indem sie die Knotentypen ändern. Weitere Informationen finden Sie unter [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn es sich bei der Engine um Valkey oder Redis Version 5.0.6 oder höher OSS handelt. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

Nachdem ein Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) erstellt wurde, kann er geändert (nach innen oder außen skaliert) werden. Weitere Informationen erhalten Sie unter [Skalierung ElastiCache](#) und [Knoten ersetzen \(Valkey und OSS Redis\)](#).

Wenn Sie einen neuen Cluster erstellen, können Sie ihn mit Daten aus dem alten Cluster bestücken, damit er nicht von Anfang an leer ist. Dieser Ansatz funktioniert nur, wenn die Cluster-Gruppe die gleiche Anzahl an Shards hat wie der alte Cluster. Dies kann hilfreich sein, wenn Sie Ihren Knotentyp oder die Engine-Version ändern müssen. Weitere Informationen erhalten Sie unter [Erstellen manueller Backups](#) und [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Mit Knoten verbinden

Verbindung zu Valkey- oder OSS Redis-Knoten herstellen

Bevor Sie versuchen, eine Verbindung zu den Valkey- oder OSS Redis-Knoten in Ihrem Cluster herzustellen, müssen Sie über die Endpunkte für die Knoten verfügen. Informationen zum Ermitteln der Endpunkte finden Sie unter:

- [Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden von Endpunkten für einen Valkey- oder Redis-Cluster OSS \(Clustermodus aktiviert\) \(Konsole\)](#)
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Endpunkte finden \(\) ElastiCache API](#)

Im folgenden Beispiel verwenden Sie das Hilfsprogramm `valkey-cli`, um eine Verbindung zu einem Cluster herzustellen, auf dem Valkey oder Redis ausgeführt wird. OSS

Note

[Weitere Informationen zu verfügbaren Befehlen finden Sie auf der Commands-Webseite.](#)

Um mit der Valkey-CLI eine Verbindung zu einem Valkey- oder OSS Redis-Cluster herzustellen

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her.

Note

Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2Erste Schritte“](#).

2. Um die GNU Compiler Collection zu erstellen `valkey-cli`, herunterzuladen und zu installieren (`gcc`). Geben Sie an der Befehlszeile Ihrer EC2 Instanz den folgenden Befehl ein und geben Sie ihn `y` an der Bestätigungsaufforderung ein.

```
sudo yum install gcc
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
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. Laden Sie das `valkey-cli` Utility herunter und kompilieren Sie es. Dieses Hilfsprogramm ist in der Valkey-Softwareverteilung enthalten. Geben Sie an der Befehlszeile Ihrer EC2 Instanz die folgenden Befehle ein:

 Note

Führen Sie bei Ubuntu-Systemen `make distclean` aus, bevor Sie `make` ausführen.

```
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. Geben Sie an der Befehlszeile Ihrer EC2 Instance den folgenden Befehl ein.

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

Eine Valkey- oder OSS Redis-Befehlszeile ähnlich der folgenden wird angezeigt.

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

5. Testen Sie die Verbindung, indem Sie die Befehle Valkey oder Redis ausführen. OSS

Sie sind jetzt mit dem Cluster verbunden und können Valkey- oder Redis-Befehle ausführen. OSS Im Folgenden finden Sie einige Beispielbefehle mit ihren Valkey- oder Redis-Antworten. 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
```

Informationen zum Herstellen von Verbindungen zu Knoten oder Clustern mit Secure Sockets Layer (SSL) -Verschlüsselung (bei der Übertragung aktiviert) finden Sie unter. [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#)

Verbindung zu Memcached-Knoten herstellen

Bevor Sie versuchen, eine Verbindung mit Ihrem Memcached-Cluster herzustellen, benötigen Sie die Endpunkte für die Knoten. Informationen zum Ermitteln der Endpunkte finden Sie unter:

- [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#)
- [Suchen von Endpunkten \(AWS CLI\)](#)

- [Endpunkte finden \(\) ElastiCache API](#)

Im folgenden Beispiel verwenden Sie das telnet-Dienstprogramm, um eine Verbindung mit einem Knoten herzustellen, der Memcached ausführt.

 Note

Weitere Informationen über Memcached und verfügbare Memcached-Befehle finden Sie auf der [Memcached-Website](#).

So stellen Sie eine Verbindung mit einem Knoten mithilfe von telnet her

1. Connect Sie mithilfe des Verbindungsdienstprogramms Ihrer Wahl eine Verbindung zu Ihrer EC2 Amazon-Instance her.

 Note

Anweisungen zum Herstellen einer Verbindung mit einer EC2 Amazon-Instance finden Sie im [Amazon-Handbuch „EC2Erste Schritte“](#).

2. Laden Sie das Telnet-Hilfsprogramm herunter und installieren Sie es auf Ihrer EC2 Amazon-Instance. Geben Sie an der Befehlszeile Ihrer EC2 Amazon-Instance den folgenden Befehl und an der Befehlszeile y ein.

```
sudo yum install telnet
```

Die Ausgabe sieht in etwa folgendermaßen aus.

```
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. Geben Sie an der Befehlszeile Ihrer EC2 Amazon-Instance den folgenden Befehl ein und ersetzen Sie den in diesem Beispiel gezeigten durch den Endpunkt Ihres Knotens.

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

Die Ausgabe sieht in etwa folgendermaßen aus.

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

4. Prüfen Sie die Verbindung mithilfe der Memcached-Befehle.

Sie sind nun mit einem Knoten verbunden und können Memcached-Befehle ausführen. Im Folgenden wird ein Beispiel gezeigt.

```
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
>
```

Unterstützte Knotentypen

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2Amazon-Instanztypen](#).

Informationen darüber, welche Knotengröße zu verwenden ist, finden Sie unter [Auswahl der Knotengröße](#).

Themen

- [Aktuelle Generation \(Memcached\)](#)
- [Aktuelle Generation \(Valkey oder Redis\) OSS](#)
- [Unterstützte Knotentypen nach AWS -Region](#)
- [Instances mit Spitzenlastleistung](#)
- [Verwandte Informationen](#)

Aktuelle Generation (Memcached)

Die folgende Tabellen zeigen die Basisbandbreite und die Burst-Bandbreite für Instance-Typen, die den Mechanismus der Netzwerk-E/A-Gutschrift verwenden, um über ihre Basisbandbreite hinaus zu gehen.

Note

Instance-Typen mit burstfähiger Netzwerkleistung nutzen ein Netzwerk-E/A-Guthabenmechanismus, um ihre Basisbandbreite nach bestem Aufwand überschreiten können.

Allgemeines

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.m7g.large		0,937	12,5
cache.m7g.xlarge		1,876	12,5
cache.m7g.2xlarge		3,75	15
cache.m7g.4xlarge		7,5	15
cache.m7g.8xlarge		15	N/A
cache.m7g.12xlarge		22,5	N/A
cache.m7g.16xlarge		30	N/A
cache.m6g.large	1,5,16	0.75	10.0
cache.m6g.xlarge	1.5,16	1,25	10.0
dache.m6g.2xlarge	1,5,16	2.5	10.0
dache.m6g.4xlarge	1.5,16	5.0	10.0
dache.m6g.8xlarge	1.5,16	12	N/A
dache.m6g.12xlarge	1.5,16	20	N/A
dache.m6g.16xlarge	1.5,16	25	N/A
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/A	N/A

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.m5.24xlarge	1.5,16	N/A	N/A
cache.m4.large	1.5,16	0,45	1.2
cache.m4.xlarge	1,5,16	0.75	2.8
cache.m4.2xlarge	1.5,16	1,0	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

Für Memcached optimierter Speicher

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r7g.large		0,937	12,5

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r7g.xlarge		1,876	12,5
cache.r7g.2xlarge		3,75	15
cache.r7g.4xlarge		7,5	15
cache.r7g.8xlarge		15	N/A
cache.r7g.12xlarge		22,5	N/A
cache.r7g.16xlarge		30	N/A
cache.r6g.large	1,5,16	0.75	10.0
cache.r6g.xlarge	1.5,16	1,25	10.0
dache.r6g.2xlarge	1,5,16	2.5	10.0
dache.r6g.4xlarge	1.5,16	5.0	10.0
dache.r6g.8xlarge	1.5,16	12	N/A
dache.r6g.12xlarge	1.5,16	20	N/A
cache.r6g.16xlarge	1.5,16	25	N/A
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/A
cache.r5.24xlarge	1.5,16	25	N/A
cache.r4.large	1.5,16	0.75	10.0

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r4.xlarge	1.5,16	1,25	10.0
cache.r4.2xlarge	1,5,16	2.5	10.0
cache.r4.4xlarge	1.5,16	5.0	10.0
cache.r4.8xlarge	1.5,16	12	N/A
cache.r4.16xlarge	1.5,16	25	N/A

Für Memcached optimiertes Netzwerk

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/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/A
cache.c7gn.8xlarge	1.6.6	100	N/A
cache.c7gn.12xlarge	1.6.6	150	N/A
cache.c7gn.16xlarge	1.6.6	200	N/A

Aktuelle Generation (Valkey oder Redis) OSS

Weitere Informationen zur vorherigen Generation finden Sie unter [Knoten der vorherigen Generation](#).

Note

Instance-Typen mit burstfähiger Netzwerkleistung nutzen ein Netzwerk-E/A-Guthabenmechanismus, um ihre Basisbandbreite nach bestem Aufwand überschreiten können.

Allgemeines

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes I/O-Multiplexing (Redis 7.0.4+) OSS	Baselinne Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/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/A
cache.m7g.12xlarge	6.2	Y	Y	Y	22,5	N/A
cache.m7g.16xlarge	6.2	Y	Y	Y	30	N/A

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes Bandbreiten-/I/O-Multiplexing (Redis 7.0.4+) OSS	Baselinne (Gbit/s)	Maximale Bandbreite (Gbit/s)
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
dache.m6g.2xlarge	5,0,6	Y	Y	Y	2.5	10.0
dache.m6g.4xlarge	5.0.6	Y	Y	Y	5.0	10.0
dache.m6g.8xlarge	5.0.6	Y	Y	Y	12	N/A
dache.m6g.12xlarge	5.0.6	Y	Y	Y	20	N/A
dache.m6g.16xlarge	5.0.6	Y	Y	Y	25	N/A
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

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes I/O-Multiplexing (Redis 7.0.4+) OSS	Baselines Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.m5.12xlarge	3.2.4	Y	Y	Y	12	N/A
cache.m5.24xlarge	3.2.4	Y	Y	Y	25	N/A
cache.m4.large	3.2.4	N	N	N	0,45	1.2
cache.m4.xlarge	3.2.4	Y	N	N	0.75	2,8
cache.m4.2xlarge	3.2.4	Y	Y	Y	1,0	10.0
cache.m4.4xlarge	3.2.4	Y	Y	Y	2.0	10.0
cache.m4.10xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.t4g.micro	3.2.4	N	N	N	0,064	5.0
cache.t4g.small	5,0,6	N	N	N	0,128	5.0
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

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes I/O-Multiplexing (Redis 7.0.4+) OSS	Baselines (Gbit/s)	Maximale Bandbreite (Gbit/s)
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

RAM-optimiert

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes I/O-Multiplexing (Redis 7.0.4+) OSS	Baselines (Gbit/s)	Maximale Bandbreite (Gbit/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

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes Bandbreiten/IO-Multiplexing (Redis 7.0.4+) OSS	Baselinbandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r7g .4xlarge	6.2	Y	Y	Y	7,5	15
cache.r7g .8xlarge	6.2	Y	Y	Y	15	N/A
cache.r7g .12xlarge	6.2	Y	Y	Y	22,5	N/A
cache.r7g .16xlarge	6.2	Y	Y	Y	30	N/A
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
dache.r6g .2xlarge	5,0,6	Y	Y	Y	2.5	10.0
dache.r6g .4xlarge	5.0.6	Y	Y	Y	5.0	10.0
dache.r6g .8xlarge	5.0.6	Y	Y	Y	12	N/A
dache.r6g .12xlarge	5.0.6	Y	Y	Y	20	N/A

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes Bandbreite I/O-Multiplexing (Redis 7.0.4+) OSS	Baselinne (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r6g.16xlarge	5.0.6	Y	Y	Y	25	N/A
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/A
cache.r5.24xlarge	3.2.4	Y	Y	Y	25	N/A
cache.r4.large	3.2.4	N	N	N	0.75	10.0
cache.r4.xlarge	3.2.4	Y	N	N	1,25	10.0
cache.r4.2xlarge	3.2.4	Y	Y	Y	2.5	10.0
cache.r4.4xlarge	3.2.4	Y	Y	Y	5.0	10.0
cache.r4.8xlarge	3.2.4	Y	Y	Y	12	N/A
cache.r4.16xlarge	3.2.4	Y	Y	Y	25	N/A

Mit Daten-Tiering optimierter Speicher

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLS Ausladen (Redis 6.2.5+) OSS	Verbessertes I/O-Multiplexing (Redis 7.0.4+) OSS	Baselines (Gbit/s)	Maximale Bandbreite (Gbit/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/A
cache.r6gd.12xlarge	6.2.0	Y	Y	Y	20	N/A
cache.r6gd.16xlarge	6.2.0	Y	Y	Y	25	N/A

Netzwerkoptimiert

Instance-Typ	Unterstützte Mindestversion von Redis OSS	Verbesserte I/O (Redis OSS 5.0.6+)	TLSAusladen (Redis 6.2.5+) OSS	Verbessertes I/O-Multiplexing (Redis 7.0.4+) OSS	Baselinbandbreite (Gbit/s)	Maximale Bandbreite (Gbit/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/A
cache.c7gn.8xlarge	6.2	Y	Y	Y	100	N/A
cache.c7gn.12xlarge	6.2	Y	Y	Y	150	N/A
cache.c7gn.16xlarge	6.2	Y	Y	Y	200	N/A

Unterstützte Knotentypen nach AWS -Region

Die unterstützten Knotentypen können je nach AWS Region variieren. Weitere Informationen finden Sie unter [ElastiCacheAmazon-Preise](#).

Instances mit Spitzenlastleistung

Sie können in Amazon Allzweck-Burstable T4g-, T3-Standard- und T2-Standard-Cache-Knoten starten. ElastiCache Diese Knoten bieten ein grundlegendes CPU Leistungsniveau mit der Möglichkeit, die CPU Auslastung jederzeit zu erhöhen, bis die angesammelten Credits aufgebraucht sind. Eine CPU Guthaben entspricht der Leistung eines kompletten CPU Kerns für eine Minute.

ElastiCacheDie T4g-, T3- und T2-Knoten von Amazon sind standardmäßig konfiguriert und eignen sich für Workloads mit einer durchschnittlichen CPU Auslastung, die durchweg unter der Basisleistung der Instance liegt. Um den Ausgangswert zu übertreffen, gibt der Node Credits aus, die er in seinem Guthaben gesammelt hat. CPU Wenn das beim Knoten angefallene Guthaben sinkt, wird die Leistung schrittweise bis auf Baseline-Leistung gesenkt. Diese schrittweise Senkung stellt sicher, dass der Knoten keinen starken Leistungseinbruch erlebt, wenn sein aufgelaufenes Guthaben CPU aufgebraucht ist. Weitere Informationen finden Sie unter [CPU Credits and Baseline Performance for Burstable Performance Instances](#) im EC2Amazon-Benutzerhandbuch.

In der folgenden Tabelle sind die Knotentypen mit Burstable Performance und die Rate aufgeführt, mit der CPU Credits pro Stunde verdient werden. Sie zeigt auch die maximale Anzahl an verdienten CPU Credits, die ein Knoten sammeln kann, sowie die Anzahl der Credits vCPUs pro Knoten. Darüber hinaus wird das Ausgangsleistungsniveau als Prozentsatz der gesamten Kernleistung (unter Verwendung eines einzigen VCPU) angegeben.

CPU pro Stunde verdiente Credits	Maximal verdiente Guthaben, die angesammelt werden können*	vCPUs	Ausgangsleistung pro V CPU	Arbeitsspeicher (GiB)	Netzwerkleistung
12	288	2	10 %	0.5	Bis zu 5 Gigabit
24	576	2	20 %	1,37	Bis zu 5 Gigabit
24	576	2	20 %	3,09	Bis zu 5 Gigabit

CPU pro Stunde verdiente Credits	Maximal verdiente Guthaben, die angesammelt werden können*	vCPUs	Ausgangsleistung pro V CPU	Arbeitsspeicher (GiB)	Netzwerkleistung
12	288	2	10 %	0,5	Bis zu 5 Gigabit
24	576	2	20 %	1,37	Bis zu 5 Gigabit
24	576	2	20 %	3,09	Bis zu 5 Gigabit
6	144	1	10 %	0,5	Gering bis mäßig
12	288	1	20 %	1,55	Gering bis mäßig
24	576	2	20 %	3,22	Gering bis mäßig

* Die Anzahl der Guthaben, die angesammelt werden können, entspricht dem Guthaben, das in einem 24-Stunden-Zeitraum verdient werden kann.

** Die Ausgangsleistung in der Tabelle bezieht sich auf einen Wert pro V. CPU. Einige Knotengrößen, die mehr als ein V haben CPU. Für diese berechnen Sie die CPU Basisauslastung für den Knoten, indem Sie den CPU V-Prozentsatz mit der Anzahl von vCPUs multiplizieren.

Die folgenden CPU Kreditkennzahlen sind für Burstable-Performance-Instances der Typen T3 und T4g verfügbar:

Note

Diese Metriken sind nicht für T2-Instances mit Spitzenleistung verfügbar.

- CPUcreditUsage
- CPUcreditBalance

[Weitere Informationen zu diesen Kennzahlen finden Sie unter CPU Kreditkennzahlen.](#)

Darüber hinaus sollten Sie Folgendes beachten:

- Alle Knotentypen der aktuellen Generation werden standardmäßig in einer virtuellen privaten Cloud (VPC) erstellt, die auf Amazon VPC basiert.
- Redis-Dateien, die OSS nur zum Anhängen dienen (AOF), werden für T2-Instances nicht unterstützt. OSSRedis-Konfigurationsvariablen `appendonly` und werden in OSS Redis-Version 2.8.22 und höher `appendfsync` nicht unterstützt.

Verwandte Informationen

- [ElastiCache Amazon-Produktmerkmale und -details](#)
- [Memcached-Knotentyp-spezifische Parameter für Memcached](#)
- [Valkey- und Redis-Parameter OSS](#)
- [Verschlüsselung bei der Übertragung TLS \(\)](#)

Knoten neu starten

Bei einigen Änderungen muss ein Redis OSS - oder Memcached-Cluster neu gestartet werden, damit die Änderungen übernommen werden. Beispielsweise wird bei einigen Parametern eine Änderung am Parameterwert in einer Parametergruppe erst nach einem Neustart übernommen.

Themen

- [Redis-Knoten neu OSS starten \(nur Clustermodus deaktiviert\)](#)
- [Einen Cluster für Memcached neu starten](#)

Redis-Knoten neu OSS starten (nur Clustermodus deaktiviert)

Für Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) lauten die Parameter in Parametergruppen, die erst nach dem Neustart angewendet werden, wie folgt:

- activerehashing
- databases

Redis-Knoten können nur über die Konsole aktualisiert werden. ElastiCache Sie können immer nur einen einzelnen Knoten neu starten. Um mehrere Knoten neu zu starten, müssen Sie den Vorgang für jeden Knoten wiederholen.

Änderungen an den Parametern Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
 - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert). Siehe [Löschen eines Clusters in ElastiCache](#).
 3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

Unter Verwendung des AWS Management Console

Sie können einen Knoten mit der ElastiCache Konsole neu starten.

So starten Sie einen Knoten (Konsole) neu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die zutreffende AWS Region aus.
3. Wählen Sie im linken Navigationsbereich Redis aus. OSS

Eine Liste von Clustern, auf denen Redis OSS ausgeführt wird, wird angezeigt.

4. Wählen Sie den Cluster unter Cluster-Name aus.
5. Unter Knotenname wählen Sie das Optionsfeld neben dem Knoten aus, den Sie neu starten möchten.
6. Wählen Sie Aktionen und dann Neustart Knoten.

Um mehrere Knoten neu zu starten, wiederholen Sie die Schritte 2 bis 5 für jeden Knoten, den Sie neu starten möchten. Es ist nicht nötig zu warten, bis ein Knoten den Neustart beendet hat, um einen anderen neu zu starten.

Einen Cluster für Memcached neu starten

Wenn Sie einen Memcached-Cluster neu starten, leert der Cluster alle seine Daten und startet seine Engine neu. Während dieses Vorgangs können Sie nicht auf den Cluster zugreifen. Da der Cluster alle seine Daten bereinigt hat, starten Sie mit einem leeren Cluster, wenn der Cluster wieder verfügbar ist.

Sie können einen Cluster mit der ElastiCache Konsole, dem oder dem neu starten. AWS CLI ElastiCache API Unabhängig davon, ob Sie die ElastiCache Konsole, die AWS CLI oder die verwenden ElastiCache API, können Sie nur den Neustart eines einzelnen Clusters initiieren. Um mehrere Cluster neu zu starten, müssen Sie den Prozess oder Vorgang iterieren.

Mit dem AWS Management Console

Sie können einen Cluster mit der ElastiCache Konsole neu starten.

So starten Sie einen Cluster neu (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.

2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, für die Sie sich interessieren.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, den Sie neu starten möchten.

Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.

4. Wählen Sie den neu zu startenden Cluster aus, indem Sie das Kontrollkästchen links neben dem Clusternamen aktivieren.

Die Schaltfläche Reboot (Neustart) wird aktiv.

Wenn Sie mehr als einen Cluster auswählen, ist das Kontrollkästchen Reboot deaktiviert.

5. Wählen Sie Reboot.

Der Bestätigungsbildschirm des Cluster-Neustarts wird angezeigt.

6. Wenn der Cluster neu gestartet werden soll, wählen Sie Reboot. Der Status des Clusters ändert sich in `rebooting cluster nodes` (Neustart von Clusterknoten).

Wenn der Cluster nicht neu gestartet werden soll, wählen Sie Cancel.

Wiederholen Sie zum Neustarten mehrerer Cluster die Schritte 2 bis 5 für jeden Cluster, den Sie neu starten möchten. Sie brauchen nicht zu warten, bis ein Cache-Cluster fertig neu gestartet wurde, bevor Sie einen anderen neu starten.

Um einen bestimmten Knoten neu zu starten, markieren Sie den Knoten und wählen Sie dann Neustart.

Verwenden Sie den AWS CLI

Verwenden Sie den `reboot-cache-cluster` CLI Vorgang, um einen Cluster (AWS CLI) neu zu starten.

Um bestimmte Knoten im Cluster neu zu starten, listen Sie mit `--cache-node-ids-to-reboot` die spezifischen Cluster auf, die neu gestartet werden sollen. Der folgende Befehl startet die Knoten 0001, 0002 und 0004 von `my-cluster` neu.

Für Linux, macOS oder Unix:

```
aws elasticache reboot-cache-cluster \
```

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

Für Windows:

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

Um alle Knoten im Cluster neu zu starten, listen Sie mit dem Parameter `--cache-node-ids-to-reboot` die Knoten-IDs aller Cluster auf. Weitere Informationen finden Sie unter [reboot-cache-cluster](#).

Mit dem ElastiCache API

Verwenden Sie die `RebootCacheCluster` Aktion ElastiCache API, um einen Cluster mit dem neu zu starten.

Um bestimmte Knoten im Cluster neu zu starten, listen Sie mit `CacheNodeIdsToReboot` die spezifischen Cluster auf, die neu gestartet werden sollen. Der folgende Befehl startet die Knoten 0001, 0002 und 0004 von `my-cluster` neu.

```
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>
```

Um alle Knoten im Cluster neu zu starten, listen Sie mit dem Parameter `CacheNodeIdsToReboot` die Knoten-IDs aller Cluster auf. Weitere Informationen finden Sie unter [RebootCacheCluster](#).

Knoten ersetzen (Valkey und OSS Redis)

Amazon aktualisiert seine Flotte ElastiCache häufig, indem Patches und Upgrades nahtlos auf Instances angewendet werden. Von Zeit zu Zeit müssen wir Ihre ElastiCache Knoten jedoch neu starten, um obligatorische Betriebssystemupdates auf den zugrunde liegenden Host anzuwenden. Diese Ersetzungen sind erforderlich, um Upgrades anzuwenden, die die Sicherheit, Zuverlässigkeit und Betriebsleistung erhöhen.

Sie haben die Möglichkeit, diese Ersetzungen vor dem geplanten Knotenersetzungsfenster jederzeit selbst zu verwalten. Wenn Sie eine Ersetzung selbst verwalten, erhält Ihre Instance das Betriebssystem-Update, wenn Sie den Knoten neu starten, und der geplante Austausch des Knotens wird abgebrochen. Sie erhalten möglicherweise weiterhin Benachrichtigungen darüber, dass der Knotenaustausch stattfinden wird. Wenn Sie die erforderliche Wartung bereits manuell vorgenommen haben, können Sie diese Benachrichtigungen ignorieren.

Note

Von Amazon automatisch generierte Ersatz-Cache-Knoten ElastiCache können unterschiedliche IP-Adressen haben. Sie sind dafür verantwortlich, Ihre Anwendungskonfiguration zu überprüfen, um sicherzustellen, dass Ihre Cache-Knoten den entsprechenden IP-Adressen zugeordnet sind.

In der folgenden Liste sind die Maßnahmen aufgeführt, die Sie ergreifen können, ElastiCache wenn Sie den Austausch eines Ihrer Valkey- oder OSS Redis-Knoten planen. Um die für Ihre Situation erforderlichen Informationen schneller zu finden, wählen Sie eine Option aus dem folgenden Menü aus.

- [Do nothing](#)— Lassen Sie Amazon den Node wie geplant ElastiCache austauschen.
- [Change your maintenance window](#) – Ändern Sie Ihr Wartungsfenster und wählen Sie einen besser geeigneten Termin.
- Konfigurationen für Valkey oder Redis OSS (Cluster-Modus aktiviert)
 - [Replace the only node in any Valkey or Redis OSS cluster](#)— Ein Verfahren zum Ersetzen eines Knotens in einem Valkey- oder OSS Redis-Cluster mithilfe von Backup und Restore.
 - [Replace a replica node in any Valkey or Redis OSS cluster](#)— Ein Verfahren zum Ersetzen einer Read-Replica in einem beliebigen Valkey- oder OSS Redis-Cluster, indem die Anzahl der Replikate erhöht oder verringert wird, ohne dass der Cluster ausfällt.

- [Replace any node in a Valkey or Redis OSS \(cluster mode enabled\) shard](#)— Ein dynamisches Verfahren ohne Cluster-Ausfallzeiten, bei dem ein Knoten in einem Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert) durch Aus- und In-Skalierung ersetzt wird.
- Konfigurationen mit Valkey oder Redis OSS (Cluster-Modus deaktiviert)
 - [Replace the only node in any Valkey or Redis OSS cluster](#)— Verfahren zum Ersetzen eines beliebigen Knotens in einem Valkey- oder OSS Redis-Cluster mithilfe von Backup und Restore.
 - [Replace a replica node in any Valkey or Redis OSS cluster](#)— Ein Verfahren zum Ersetzen einer Read-Replica in einem beliebigen Valkey- oder OSS Redis-Cluster, indem die Anzahl der Replikate erhöht oder verringert wird, ohne dass der Cluster ausfällt.
 - [Replace a node in a Valkey or Redis OSS \(cluster mode disabled\) cluster](#)— Verfahren zum Ersetzen eines Knotens in einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mithilfe von Replikation.
 - [Replace a Valkey or Redis OSS \(cluster mode disabled\) read-replica](#)— Ein Verfahren zum manuellen Ersetzen einer Read-Replica in einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert).
 - [Replace a Valkey or Redis OSS \(cluster mode disabled\) primary node](#)— Ein Verfahren zum manuellen Ersetzen des Primärknotens in einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert).

Optionen für den Austausch von Valkey- und OSS Redis-Knoten

- Nichts tun — Wenn Sie nichts tun, wird der Knoten wie geplant ElastiCache ersetzt.

Bei Nicht-Cluster-Konfigurationen mit aktiviertem Autofailover werden Cluster auf Valkey 7.2 und höher und Redis OSS 5.0.6 und höher vollständig ersetzt, während der Cluster weiterhin online bleibt und eingehende Schreib Anforderungen bearbeitet. Bei Clustern mit aktiviertem Auto Failover auf Redis OSS 4.0.10 oder niedriger stellen Sie möglicherweise eine kurze Schreibunterbrechung von bis zu einigen Sekunden im Zusammenhang mit Updates fest. DNS

Wenn der Knoten Mitglied eines Clusters ist, für das Auto Failover aktiviert ist, OSS bietet Valkey oder Redis eine verbesserte Verfügbarkeit beim Patchen, ElastiCache bei Updates und anderen wartungsbedingten Node-Austauschen.

ElastiCache Bei Valkey- oder OSS Redis-Cluster-Konfigurationen, die für die Verwendung ElastiCache mit Valkey- oder Redis-Cluster-Clients eingerichtet sind, wird der Austausch jetzt abgeschlossen, während der OSS Cluster eingehende Schreibanforderungen bearbeitet.

Bei Nicht-Cluster-Konfigurationen mit aktiviertem Autofailover werden Cluster auf Valkey 7.2 und höher und Redis OSS 5.0.6 und höher vollständig ersetzt, während der Cluster weiterhin online bleibt und eingehende Schreibanforderungen bearbeitet. Bei Clustern mit aktiviertem Auto Failover auf Redis OSS 4.0.10 oder niedriger stellen Sie möglicherweise eine kurze Schreibunterbrechung von bis zu einigen Sekunden im Zusammenhang mit Updates fest. DNS

Wenn der Knoten eigenständig ist, startet Amazon ElastiCache zuerst einen Ersatzknoten und synchronisiert dann vom vorhandenen Knoten aus. Der vorhandene Knoten ist während dieser Zeit nicht für Serviceanfragen verfügbar. Sobald die Synchronisierung abgeschlossen ist, wird der bestehende Knoten beendet und der neue Knoten nimmt seinen Platz ein. ElastiCache bemüht sich nach besten Kräften, Ihre Daten während dieses Vorgangs aufzubewahren.

- Ändern Sie Ihr Wartungsfenster — Bei geplanten Wartungsereignissen erhalten Sie eine E-Mail oder eine Benachrichtigung von ElastiCache. Wenn Sie in diesen Fällen Ihr Wartungsfenster vor der geplanten Ersatzzeit ändern, wird Ihr Knoten zur neuen Uhrzeit ersetzt. Weitere Informationen finden Sie hier:
 - [Einen ElastiCache Cluster ändern](#)
 - [Ändern einer Replikationsgruppe](#)

Note

Die Möglichkeit, Ihr Ersatzfenster zu ändern, indem Sie Ihr Wartungsfenster verschieben, ist nur verfügbar, wenn die ElastiCache Benachrichtigung ein Wartungsfenster beinhaltet. Wenn die Benachrichtigung kein Wartungsfenster enthält, können Sie Ihr Ersatzfenster nicht ändern.

Es ist beispielsweise Donnerstag, der 9. November, um 15:00 Uhr und das nächste Wartungsfenster ist am Freitag, dem 10. November, um 17:00 Uhr. Es folgen drei Szenarien mit den jeweiligen Ergebnissen:

- Sie ändern Ihr Wartungsfenster auf Freitag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben, aber vor dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Freitag, dem 10. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Samstag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben und nach dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Samstag, dem 11. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Mittwoch, 16:00 Uhr, früher in der Woche, als die aktuellen Datums- und Uhrzeitangaben. Dieser Knoten wird am kommenden Mittwoch, dem 15. November, um 16:00 Uhr ersetzt.

Detaillierte Anweisungen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

- Ersetzen Sie den einzigen Knoten in einem Valkey- oder OSS Redis-Cluster — Wenn der Cluster keine Read Replicas hat, können Sie den Knoten mit dem folgenden Verfahren ersetzen.

So ersetzen Sie den einzigen Knoten mithilfe von Sicherung und Wiederherstellung

1. Erstellen Sie einen Snapshot des Clusters des Knotens. Detaillierte Anweisungen finden Sie unter [Erstellen manueller Backups](#).
 2. Erstellen Sie einen neuen Cluster und übernehmen Sie die Daten mithilfe des Snapshots. Detaillierte Anweisungen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
 3. Löschen Sie den Cluster mit dem für das Ersetzen geplanten Knoten. Detaillierte Anweisungen finden Sie unter [Löschen eines Clusters in ElastiCache](#).
 4. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.
- Ersetzen Sie einen Replikatknoten in einem beliebigen Valkey- oder OSS Redis-Cluster — Um einen Replikatcluster zu ersetzen, erhöhen Sie die Anzahl der Replikate. Fügen Sie dafür ein

Replikate hinzu und verringern Sie die Replikanzahl, indem Sie das Replikate entfernen, das Sie ersetzen möchten. Dieser Vorgang ist dynamisch und ist nicht mit Cluster-Ausfallzeiten verbunden.

 Note

Wenn Ihr Shard oder Ihre Replikationsgruppe bereits fünf Replikate hat, kehren Sie die Schritte 1 und 2 um.

Um ein Replikate in einem beliebigen Valkey- oder Redis-Cluster zu ersetzen OSS

1. Erhöhen Sie die Replikanzahl, indem Sie ein Replikate zum Shard oder zur Replikationsgruppe hinzufügen. Weitere Informationen finden Sie unter [Erhöhen der Anzahl der Replikate in einem Shard](#).
 2. Löschen Sie das Replikate, das Sie ersetzen möchten. Weitere Informationen finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#).
 3. Aktualisieren Sie die Endpunkte in Ihrer Anwendung.
- Ersetzen Sie einen beliebigen Knoten in einem Valkey- oder Redis-Shard OSS (Clustermodus aktiviert) — Verwenden Sie Online-Resharding, um den Knoten in einem Cluster ohne Ausfallzeiten zu ersetzen. Fügen Sie zunächst einen Shard durch Ausskalieren hinzu und löschen Sie den Shard mit dem Knoten, der durch Einkalieren ersetzt werden soll.

Um einen beliebigen Knoten in einem Valkey- oder OSS Redis-Cluster (Clustermodus aktiviert) zu ersetzen

1. Horizontales Skalieren: Fügen Sie einen zusätzlichen Shard mit derselben Konfiguration wie die des bestehenden Shards mit dem zu ersetzenden Knoten hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Shards mit Online-Resharding](#).
2. Vertikales Skalieren: Löschen Sie den Shard mit dem zu ersetzenden Knoten. Weitere Informationen finden Sie unter [Entfernen von Shards mit Online-Resharding](#).
3. Aktualisieren Sie die Endpunkte in Ihrer Anwendung.

- Ersetzen Sie einen Knoten in einem Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) — Wenn es sich bei dem Cluster um einen Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) ohne Read Replicas handelt, verwenden Sie das folgende Verfahren, um den Knoten zu ersetzen.

So ersetzen Sie den Knoten mithilfe der Replikation (nur mit deaktiviertem Cluster-Modus)

1. Fügen Sie dem Cluster die Replikation mit dem Knoten hinzu, der als primärer Knoten ersetzt werden soll. Aktivieren Sie Multi-AZ auf diesem Cluster nicht. Detaillierte Anweisungen finden Sie unter [Um die Replikation zu einem Valkey- oder OSS Redis-Cluster ohne Shards hinzuzufügen](#).
 2. Fügen Sie dem Cluster ein Lesereplikat hinzu. Detaillierte Anweisungen finden Sie unter [Um Knoten zu einem ElastiCache Cluster hinzuzufügen \(Konsole\)](#).
 3. Stufen Sie das soeben erstellte Lesereplikat auf Primär hoch. Detaillierte Anweisungen finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey- oder Redis-Replikationsgruppen OSS \(Clustermodus deaktiviert\)](#).
 4. Löschen Sie den für das Ersetzen geplanten Knoten. Detaillierte Anweisungen finden Sie unter [Knoten aus einem ElastiCache Cluster entfernen](#).
 5. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.
- Ersetzen Sie eine Valkey- oder Redis-Lese-Replik OSS (Cluster-Modus deaktiviert) — Wenn es sich bei dem Knoten um eine Read-Replica handelt, ersetzen Sie den Knoten.

Wenn Ihr Cluster nur einen Replikatknoten enthält und Multi-AZ aktiviert ist, müssen Sie Multi-AZ deaktivieren, bevor Sie das Replikat löschen können. Detaillierte Anweisungen finden Sie unter [Ändern einer Replikationsgruppe](#).

Um ein Valkey- oder Redis-Lese-Replikat (Cluster-Modus deaktiviert) zu ersetzen OSS

1. Löschen Sie das für den Austausch geplante Replikat. Detaillierte Informationen finden Sie hier:
 - [Verringern der Anzahl der Replikate in einem Shard](#)
 - [Knoten aus einem ElastiCache Cluster entfernen](#)

2. Fügen Sie ein neues Replikate hinzu, das das auszutauschende Replikate ersetzen soll. Wenn Sie denselben Namen wie für das soeben gelöschte Replikate verwenden, können Sie Schritt 3 überspringen. Detaillierte Informationen finden Sie hier:
 - [Erhöhen der Anzahl der Replikate in einem Shard](#)
 - [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#)
 3. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Replikats.
 4. Wenn Sie die Multi-AZ-Funktion beim Start deaktiviert haben, aktivieren Sie sie nun wieder. Detaillierte Anweisungen finden Sie unter [Aktivieren von Multi-AZ](#).
-
- Einen Valkey- oder Redis-Primärknoten OSS (Cluster-Modus deaktiviert) ersetzen — Wenn es sich bei dem Knoten um den primären Knoten handelt, stufen Sie zuerst eine Read-Replica zum Primärknoten herauf. Löschen Sie dann das Replikate, das vorher der primäre Knoten war.

Wenn Ihr Cluster nur ein Replikate enthält und Multi-AZ aktiviert ist, müssen Sie Multi-AZ deaktivieren, bevor Sie das Replikate in Schritt 2 löschen können. Detaillierte Anweisungen finden Sie unter [Ändern einer Replikationsgruppe](#).

Um einen Valkey- oder OSS Redis-Primärknoten (Cluster-Modus deaktiviert) zu ersetzen

1. Stufen Sie ein Lesereplikate auf Primär hoch. Detaillierte Anweisungen finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey- oder Redis-Replikationsgruppen OSS \(Clustermodus deaktiviert\)](#).
2. Löschen Sie den Knoten, der ausgetauscht werden soll (der alte primäre Knoten). Detaillierte Anweisungen finden Sie unter [Knoten aus einem ElastiCache Cluster entfernen](#).
3. Fügen Sie ein neues Replikate hinzu, das das auszutauschende Replikate ersetzen soll. Wenn Sie denselben Namen wie den des soeben gelöschten Knotens verwenden, können Sie das Ändern der Endpunkte in Ihrer Anwendung überspringen.

Detaillierte Anweisungen finden Sie unter [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

4. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.

5. Wenn Sie die Multi-AZ-Funktion beim Start deaktiviert haben, aktivieren Sie sie nun wieder. Detaillierte Anweisungen finden Sie unter [Aktivieren von Multi-AZ](#).

Knoten ersetzen (Memcached)

Amazon ElastiCache (Memcached) aktualisiert seine Flotte häufig, indem Patches und Upgrades nahtlos auf Instances angewendet werden. Von Zeit zu Zeit müssen wir jedoch Ihre ElastiCache (Memcached-) Nodes neu starten, um obligatorische Betriebssystem-Updates auf den zugrunde liegenden Host anzuwenden. Diese Ersetzungen sind erforderlich, um Upgrades anzuwenden, die die Sicherheit, Zuverlässigkeit und Betriebsleistung erhöhen.

Sie haben die Möglichkeit, diese Ersetzungen vor dem geplanten Knotenersetzungsfenster jederzeit selbst zu verwalten. Wenn Sie eine Ersetzung selbst verwalten, erhält Ihre Instance das Betriebssystem-Update, wenn Sie den Knoten neu starten, und der geplante Austausch des Knotens wird abgebrochen. Sie erhalten möglicherweise weiterhin Benachrichtigungen darüber, dass der Knotenaustausch stattfindet. Wenn Sie die erforderliche Wartung bereits manuell vorgenommen haben, können Sie diese Benachrichtigungen ignorieren.

Note

Von Amazon automatisch generierte Ersatz-Cache-Knoten ElastiCache können unterschiedliche IP-Adressen haben. Sie sind dafür verantwortlich, Ihre Anwendungskonfiguration zu überprüfen, um sicherzustellen, dass Ihre Cache-Knoten den entsprechenden IP-Adressen zugeordnet sind.

In der folgenden Liste sind die Maßnahmen aufgeführt, die Sie ergreifen können, ElastiCache wenn Sie den Austausch eines Ihrer Memcached-Knoten planen.

- Nichts tun — Wenn Sie nichts tun, wird der Knoten wie geplant ElastiCache ersetzt. Wenn der Knoten ElastiCache automatisch durch einen neuen Knoten ersetzt wird, ist der neue Knoten zunächst leer.
- Ändern Sie Ihr Wartungsfenster — Bei geplanten Wartungsereignissen erhalten Sie eine E-Mail oder eine Benachrichtigung von ElastiCache. Wenn Sie in diesen Fällen Ihr Wartungsfenster vor der geplanten Ersatzzeit ändern, wird Ihr Knoten zur neuen Uhrzeit ersetzt. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

Note

Die Möglichkeit, Ihr Ersatzfenster zu ändern, indem Sie Ihr Wartungsfenster verschieben, ist nur verfügbar, wenn die ElastiCache Benachrichtigung ein Wartungsfenster beinhaltet.

Wenn die Benachrichtigung kein Wartungsfenster enthält, können Sie Ihr Ersatzfenster nicht ändern.

Es ist beispielsweise Donnerstag, der 9. November, um 15:00 Uhr und das nächste Wartungsfenster ist am Freitag, dem 10. November, um 17:00 Uhr. Es folgen drei Szenarien mit den jeweiligen Ergebnissen:

- Sie ändern Ihr Wartungsfenster auf Freitag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben, aber vor dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Freitag, dem 10. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Samstag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben und nach dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Samstag, dem 11. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Mittwoch, 16:00 Uhr, früher in der Woche, als die aktuellen Datums- und Uhrzeitangaben. Dieser Knoten wird am kommenden Mittwoch, dem 15. November, um 16:00 Uhr ersetzt.

Detaillierte Anweisungen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

- Manuelles Ersetzen des Knotens – Wenn Sie den Knoten vor dem nächsten Wartungsfenster ersetzen müssen, ersetzen Sie den Knoten manuell.

Wenn Sie den Knoten manuell ersetzen, werden Schlüssel umverteilt. Die Umverteilung verursacht Cache-Fehler.

So ersetzen Sie einen Memcached-Knoten manuell

1. Löschen Sie den für das Ersetzen geplanten Knoten. Detaillierte Anweisungen finden Sie unter [Knoten aus einem ElastiCache Cluster entfernen](#).
2. Fügen Sie dem Cluster einen neuen Knoten hinzu. Detaillierte Anweisungen finden Sie unter [Hinzufügen von Knoten zu einem ElastiCache Cluster](#).
3. Wenn Sie Auto Discovery in diesem Cluster nicht verwenden, rufen Sie Ihre Anwendung auf und tauschen Sie jede Instance des Endpunkts des alten mit dem Endpunkt des neuen Knotens aus.

Reservierte Knoten

Die Reservierung eines oder mehrerer ElastiCache Knoten kann eine Möglichkeit für Sie sein, die Kosten zu senken. Für reservierte Knoten wird eine Vorausgebühr erhoben, die von der Art des Knotens und der Dauer der Reservierung – ein oder drei Jahre – abhängt.

Um festzustellen, ob reservierte Knoten für Ihre Anwendungsfälle Kosten einsparen, ermitteln Sie zunächst die Knotengröße und wie viele Knoten Sie benötigen. Schätzen Sie dann die Verwendung des Knotens und vergleichen Sie die Gesamtkosten von On-Demand-Knoten im Gegensatz zu reservierten Knoten. Sie können die Nutzung von reservierten und On-Demand-Knoten in Ihren Clustern mischen und anpassen. Preisinformationen finden Sie unter [ElastiCache Amazon-Preise](#).

Themen

- [Preismanagement mit reservierten Knotenpunkten](#)
- [Standardangebote für reservierte Knoten](#)
- [Größe: flexible reservierte Knoten](#)
- [Löschen eines reservierten Knotens](#)
- [Ältere Angebote für reservierte Knoten](#)
- [Abrufen von Informationen über Angebote für reservierte Knoten](#)
- [Kauf eines reservierten Knotens](#)
- [Abrufen von Informationen über Ihre reservierten Knoten](#)

Preismanagement mit reservierten Knotenpunkten

Durch Reservieren eines oder mehrerer Knoten lassen sich die Kosten möglicherweise reduzieren. Für reservierte Knoten wird eine Vorausgebühr erhoben, die von der Art des Knotens und der Dauer der Reservierung – ein oder drei Jahre – abhängt. Diese Gebühr ist wesentlich geringer als die stündliche Nutzungsgebühr, die bei On-Demand-Knoten anfällt.

Um festzustellen, ob reservierte Knoten für Ihre Anwendungsfälle Kosten einsparen, ermitteln Sie zunächst die Knotengröße und wie viele Knoten Sie benötigen. Schätzen Sie dann die Nutzung des Knotens und vergleichen Sie die Gesamtpreise, die Ihnen durch die Nutzung von On-Demand-Knoten im Vergleich zu reservierten Knoten entstehen. Sie können die Nutzung von reservierten und On-Demand-Knoten in Ihren Clustern mischen und anpassen. Preisinformationen finden Sie unter [ElastiCache Amazon-Preise](#).

AWS Region, Knotentyp und Laufzeit müssen beim Kauf ausgewählt werden und können später nicht mehr geändert werden.

Sie können das AWS Management Console, das oder das verwenden AWS CLI, ElastiCache API um verfügbare Angebote für reservierte Knoten aufzulisten und zu erwerben.

Weitere Informationen zu reservierten Knoten finden Sie unter [Amazon ElastiCache Reserved Nodes](#).

Standardangebote für reservierte Knoten

Wenn Sie eine Reserved Node Instance (RI) bei Amazon erwerben ElastiCache, können Sie sich verpflichten, für die Dauer der Reserved Node Instance einen vergünstigten Tarif für einen bestimmten Node-Instance-Typ und eine AWS Region zu erhalten. Um eine Amazon ElastiCache Reserved Node Instance zu verwenden, erstellen Sie eine neue ElastiCache Node-Instance, genau wie bei einer On-Demand-Instance.

Wenn die Spezifikationen der neuen Reserve-Node-Instance mit einer bestehenden Reserved Node-Instance für Ihr Konto übereinstimmen, wird Ihnen der für die Reserved Node-Instance angebotene vergünstigte Tarif in Rechnung gestellt. Andernfalls wird die Knoten-Instance auf Basis eines On-Demand-Preises abgerechnet. Diese Standardversionen RIs sind ab den Instance-Familien R5 und M5 verfügbar.

Note

Alle im Folgenden erörterten Angebotstypen sind mit Laufzeiten von einem Jahr und drei Jahren erhältlich.

Angebotstypen

No Upfront RI ermöglicht den Zugriff auf eine Reserved ElastiCache Instance, ohne dass eine Vorauszahlung erforderlich ist. Ihre No Upfront Reserved ElastiCache Instance berechnet für jede Stunde innerhalb der Laufzeit einen vergünstigten Stundensatz, unabhängig von der Nutzung.

Bei teilweiser Vorauszahlung muss ein Teil der Reserved ElastiCache Instance im Voraus bezahlt werden. Die innerhalb der Laufzeit verbleibenden Stunden werden unabhängig von der Nutzung zu einem vergünstigten Stundensatz berechnet. Diese Option ersetzt die Legacy-Option Hohe Auslastung, die im nächsten Abschnitt erklärt wird.

Vorauszahlung RI muss der gesamte Betrag zu Beginn der Laufzeit der RI bezahlt werden. Für den Rest der Laufzeit entstehen Ihnen keine weiteren Preise, unabhängig von der Anzahl der genutzten Stunden.

Größe: flexible reservierte Knoten

Alle reservierten Knoten sind größenflexibel. Wenn Sie einen reservierten Knoten kaufen, geben Sie unter anderem den Knotentyp an, zum Beispiel `cache.r6g.xlarge`. Weitere Informationen zu Knotentypen finden Sie unter [ElastiCacheAmazon-Preise](#).

Wenn Sie über einen Node verfügen und diesen auf eine größere Kapazität skalieren müssen, wird Ihr reservierter Node automatisch auf Ihren skalierten Node angewendet. Das heißt, Ihre reservierten Knoten werden automatisch für die Nutzung beliebiger Größe in derselben Knotenfamilie verwendet. Größenflexible reservierte Knoten sind für Knoten mit derselben AWS Region verfügbar. Größenflexible reservierte Knoten können nur innerhalb ihrer Knotenfamilien skaliert werden. Beispielsweise kann ein reservierter Knoten für `cache.r6g.xlarge` für `cache.r6g.2xlarge` gelten, aber nicht für `cache.r6gd.large`, da `cache.r6g` und `cache.r6gd` unterschiedliche Knotenfamilien sind.

Größenflexibilität bedeutet, dass Sie sich frei zwischen Konfigurationen innerhalb derselben Knotenfamilie bewegen können. Sie können beispielsweise ohne zusätzliche Kosten von einem reservierten `r6g.xlarge`-Knoten (8 normalisierte Einheiten) zu zwei reservierten `r6g.large`-Knoten (8 normalisierte Einheiten) ($2 \times 4 = 8$ normalisierte Einheiten) in derselben Region wechseln. AWS

Upgrade OSS reservierter Knoten von Redis auf Valkey

Mit dem Start von Valkey können Sie jetzt Ihren Redis-Rabatt für OSS reservierte Knoten auf die Valkey-Cache-Engine anwenden. ElastiCache Sie können ein Upgrade von Redis OSS auf Valkey durchführen und gleichzeitig von bestehenden Verträgen und Reservierungen profitieren. Sie können Ihre Vorteile nicht nur innerhalb der Cache-Node-Familie und -Engine nutzen, sondern auch einen größeren Mehrwert erzielen. Valkey ist im Vergleich zu Redis mit einem discount von 20% erhältlich OSS, und dank der Flexibilität für reservierte Knoten können Sie Ihre OSS reservierten Redis-Nodes verwenden, um 20% mehr laufende Valkey-Knoten abzudecken.

Zur Berechnung des ermäßigten Tarifs hat jede Kombination aus ElastiCache Knoten und Motor einen Normalisierungsfaktor, der in Einheiten gemessen wird. Reservierte Knoteneinheiten können auf jeden laufenden Knoten innerhalb der Instanzfamilie des reservierten Knotens für eine bestimmte Engine angewendet werden. OSSReservierte Redis-Nodes können zusätzlich Engine-übergreifend eingesetzt werden, um laufende Valkey-Knoten abzudecken. Da Valkey im Vergleich zu Redis OSS und Memcached günstiger ist, sind die Einheiten für einen bestimmten Instance-Typ niedriger, sodass ein OSS reservierter Redis-Node mehr Valkey-Knoten abdecken kann.

Nehmen wir als Beispiel an, Sie haben einen reservierten Knoten für einen cache.r7g.4xlarge für die OSS Redis-Engine (32 Einheiten) gekauft und betreiben einen cache.r7g.4xlarge Redis-Knoten (32 Einheiten). OSS Wenn Sie den Knoten auf Valkey aktualisieren, sinkt der Normalisierungsfaktor des laufenden Knotens auf 25,6 Einheiten, und Ihr vorhandener reservierter Knoten bietet Ihnen zusätzliche 6,4 Einheiten, die Sie gegen jeden anderen laufenden Valkey- oder OSS Redis-Knoten innerhalb der cache.r7g-Familie in der Region verwenden können. Sie könnten damit 25% eines anderen cache.r7g.4xlarge Valkey-Knotens im Konto (25,6 Einheiten) oder 100% eines cache.r7g.xlarge Valkey-Knotens (6,4 Einheiten) abdecken.

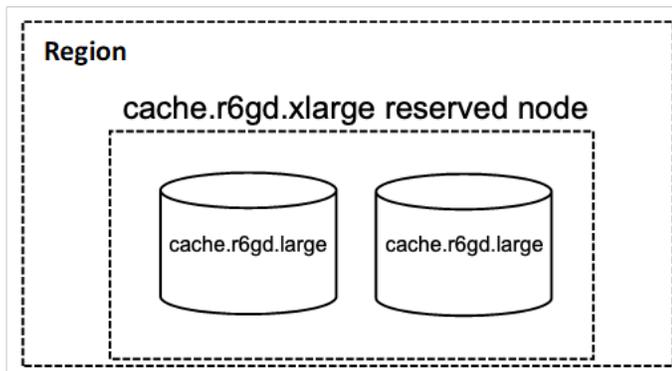
Vergleich der Nutzung mit normalisierten Einheiten

Sie können die Nutzung für verschiedene reservierte Knotengrößen vergleichen, indem Sie normalisierte Einheiten verwenden. Beispielsweise entspricht eine Nutzungsstunde auf zwei cache.r6g.4xlarge-Knoten 16 Stunden Nutzung auf einem cache.r6g.large. Die folgende Tabelle zeigt die Anzahl der normalisierten Einheiten für jede Knotengröße:

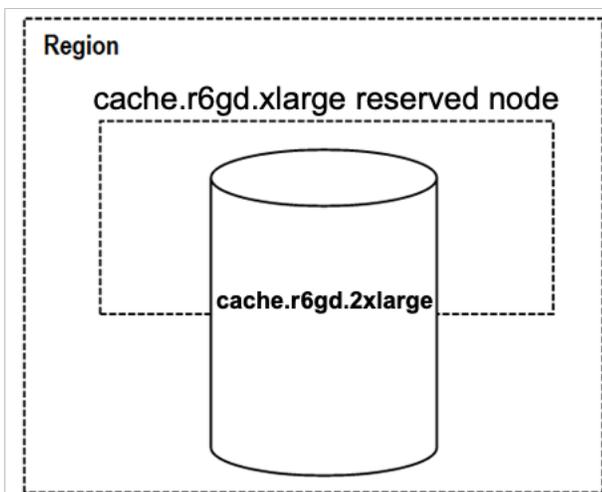
Knotengröße	Normalisierte Einheiten (RedisOSS, Memcached)	Normalisierte Einheiten (Valkey)
Micro	0.5	0.4
small	1	8.
Medium	2	1,6
large	4	3.2
xlarge	8	6.4
2xlarge	16	12.8
4xlarge	32	25,6
6xlarge	48	38,4
8xlarge	64	51,2
10xlarge	80	64
12xlarge	96	76,8

Knotengröße	Normalisierte Einheiten (RedisOSS, Memcached)	Normalisierte Einheiten (Valkey)
16xlarge	128	102,4
24xlarge	192	153,6

Beispiel: Sie kaufen einen reservierten Cache.r6gd.xlarge-Node und Sie haben zwei laufende reservierte Cache.r6gd.large-Nodes in Ihrem Konto in derselben Region. AWS In diesem Fall wird der Abrechnungsvorteil vollständig auf beide Knoten angewendet.



Wenn in Ihrem Konto in derselben AWS Region eine `cache.r6gd.2xlarge`-Instance ausgeführt wird, wird der Abrechnungsvorteil alternativ auf 50 Prozent der Nutzung des reservierten Knotens angerechnet.



Löschen eines reservierten Knotens

Die Bedingungen für einen reservierten Knoten beinhalten eine Laufzeit von einem oder drei Jahren. Sie können einen reservierten Knoten nicht kündigen. Sie können jedoch einen Knoten löschen, für den ein discount für reservierte Knoten gilt. Der Vorgang zum Löschen eines Knotens, für den ein discount für reservierte Knoten gilt, ist derselbe wie für jeden anderen Knoten.

Wenn Sie einen Knoten löschen, für den ein discount für reservierte Knoten gilt, können Sie einen anderen Knoten mit kompatiblen Spezifikationen starten. In diesem Fall erhalten Sie den Rabatt während des Reservierungszeitraums (ein Jahr oder drei Jahre).

Ältere Angebote für reservierte Knoten

Es gibt drei Stufen von Legacy-Knoten-Reservierungen – schwere Auslastung, mittlere Auslastung und leichte Auslastung. Knoten können in jeder Auslastung für ein oder drei Jahre reserviert werden. Der Knotentyp, der Auslastungsgrad und die Reservierungsdauer beeinflussen Ihre Gesamtpreise. Überprüfen Sie die Einsparungen, die reservierte Knoten Ihrem Unternehmen bringen können, indem Sie verschiedene Modelle vergleichen, bevor Sie reservierte Knoten kaufen.

Die Auslastungsstufe oder Reservierungsdauer eines erworbenen Knotens lässt sich nicht mehr ändern.

Auslastungsstufen

Reservierte Knoten mit starker Auslastung ermöglichen Workloads mit einer konsistenten Kapazitätsgrundlage oder konstante Workloads. Reservierte Knoten mit starker Auslastung erfordern zwar eine hohe Vorabzahlung, doch wenn Sie planen, mehr als 79 % der Reservierungsdauer zu nutzen, erzielen Sie die größten Einsparungen (bis zu 70 % weniger als der Preis von On-Demand-Knoten). Bei reservierten Knoten mit hoher Auslastung zahlen Sie eine einmalige Gebühr. Danach wird für die Dauer der Laufzeit eine niedrigere Stundengebühr fällig, unabhängig davon, ob Ihr Knoten läuft.

Reservierte Knoten mit mittlerer Auslastung sind die beste Option, wenn Sie vorhaben, Ihre reservierten Knoten einen Großteil der Zeit zu nutzen, und Sie entweder eine niedrigere Einmalgebühr wünschen oder nicht mehr für Ihren Knoten zahlen möchten, wenn Sie ihn abschalten. Reservierte Knoten mit mittlerer Auslastung stellen eine kosteneffiziente Option dar, wenn Sie planen, mehr als 40 % der Knotenreservierungsdauer in Anspruch zu nehmen. Mit dieser Option können Sie bis zu 64 % gegenüber dem Preis für On-Demand-Knoten sparen. Für reservierte Knoten mit mittlerer Auslastung zahlen Sie eine etwas höhere Einmalgebühr als mit reservierten Knoten mit

geringer Auslastung und profitieren von niedrigeren Nutzungstarifen pro Stunde, wenn Sie einen Knoten ausführen.

Reservierte Knoten mit geringer Auslastung sind ideal für periodische Workloads, die nur wenige Stunden pro Tag oder an wenigen Tagen pro Woche ausgeführt werden. Für reservierte Knoten mit geringer Auslastung zahlen Sie eine Einmalgebühr sowie eine ermäßigte Nutzungsgebühr pro Stunde, wenn Ihr Knoten ausgeführt wird. Sie können mit dem Sparen beginnen, wenn Ihr Knoten zu mehr als 17 % der reservierten Knotenlaufzeit läuft. Sie können während der gesamten Laufzeit Ihres reservierten Knotens bis zu 56 % gegenüber den On-Demand-Tarifen sparen.

Ältere Angebote für reservierte Knoten

Angebot	Vorabkosten	Nutzungsgebühr	Vorteil
Hohe Auslastung	Am höchsten	Niedrigste Gebühr pro Stunde. Gilt für die gesamte Dauer, unabhängig davon, ob Sie den reservierten Knoten in Anspruch nehmen.	Niedrigste Gesamtkosten, wenn Sie planen, Ihre reservierten Knoten mehr als 79 % eines 3-Jahres-Zeitraums auszuführen.
Mittlere Auslastung	Mittelschwer	Nutzungsgebühr pro Stunde für jede Stunde, die der Knoten ausgeführt wird. Keine Nutzungsgebühr pro Stunde, wenn der Knoten nicht ausgeführt wird.	Geeignet für elastische Workloads oder bei erwarteter mittlerer Nutzung, mehr als 40 % eines 3-Jahres-Zeitraums.
Leichte Auslastung	Am niedrigsten	Nutzungsgebühr pro Stunde für jede Stunde, die der Knoten ausgeführt wird. Keine	Die höchsten Gesamtpreise, wenn Sie planen, die ganze Zeit in Betrieb zu sein. Dies sind

Angebot	Vorabkosten	Nutzungsgebühr	Vorteil
		Nutzungsgebühr pro Stunde, wenn der Knoten nicht ausgeführt wird. Höchste Gebühren pro Stunde aller Angebotstarife, die Gebühren fallen jedoch nur an, wenn der reservierte Knoten ausgeführt wird.	jedoch die niedrigsten Gesamtpreise, wenn Sie planen, Ihren reservierten Knotenpunkt nur unregelmässig zu nutzen, d. h. mehr als etwa 15 % einer dreijährigen Laufzeit.
On-Demand-Verwendung (Keine reservierten Knoten)	None	Höchste Gebühr pro Stunde. Fällt nur immer dann an, wenn der Knoten ausgeführt wird.	Höchste Kosten pro Stunde.

Weitere Informationen finden Sie unter [ElastiCache Amazon-Preise](#).

Abrufen von Informationen über Angebote für reservierte Knoten

Vor dem Kauf von reservierten Knoten können Sie Informationen zu verfügbaren Angeboten für reservierte Knoten erhalten.

Die folgenden Beispiele zeigen, wie Sie Preise und Informationen zu verfügbaren Reserved Node-Angeboten mithilfe von AWS Management Console AWS CLI, und abrufen können ElastiCache API.

Themen

- [Abrufen von Informationen über Angebote für reservierte Knoten \(Konsole\)](#)
- [Abrufen von Informationen über Angebote für reservierte Knoten \(AWS CLI\)](#)
- [Informationen über Angebote für reservierte Knoten abrufen \(\) ElastiCache API](#)

Abrufen von Informationen über Angebote für reservierte Knoten (Konsole)

Verwenden Sie das folgende Verfahren, um Preise und andere Informationen zu verfügbaren reservierten Cluster-Angeboten mit dem zu erhalten. AWS Management Console

So erhalten Sie Informationen zu verfügbaren Angeboten für reservierte Knoten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Reserved Nodes (Reserved Nodes) aus.
3. Wählen Sie Purchase Reserved Node (Reservierten Knoten kaufen) aus.
4. Wählen Sie für Engine entweder Valkey, Redis OSS oder Memcached aus.
5. Treffen Sie zur Ermittlung der verfügbaren Angebote eine Auswahl für die folgenden Optionen:
 - Node Type
 - Laufzeit
 - Angebotstyp

Nachdem Sie Ihre Auswahl getroffen haben, werden die Kosten pro Knoten und die Gesamtknoten der ausgewählten Optionen unter Reservation details (Reservierungsdetails) angezeigt.

6. Wählen Sie Cancel, um den Kaufvorgang für diese Knoten abubrechen und Kosten zu vermeiden.

Abrufen von Informationen über Angebote für reservierte Knoten (AWS CLI)

Um Preise und andere Informationen zu verfügbaren Angeboten für reservierte Knoten für Valkey oder Redis zu erhalten, geben Sie an der Befehlszeile den folgenden Befehl ein:

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

Dieser Vorgang erzeugt eine Ausgabe, die der folgenden ähnelt (JSONFormat):

```
{
  "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"
      }
    ]
  }
}

```

Um Preise und andere Informationen zu verfügbaren Reserved Node-Angeboten für Memcached zu erhalten, geben Sie an der Befehlszeile den folgenden Befehl ein:

```

{
  "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"
      }
    ]
  }
}
```

Weitere Informationen finden Sie in der Referenz unter [describe-reserved-cache-nodes-offers](#). AWS CLI

Informationen über Angebote für reservierte Knoten abrufen () ElastiCache API

Um Preisinformationen und Informationen zu verfügbaren Angeboten für reservierte Knoten zu erhalten, rufen Sie die DescribeReservedCacheNodesOfferings-Aktion auf.

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>
```

Weitere Informationen finden Sie [DescribeReservedCacheNodesOfferings](#) in der ElastiCache API Referenz.

Kauf eines reservierten Knotens

Die folgenden Beispiele zeigen, wie Sie ein Angebot für reservierte Knoten mithilfe der AWS Management Console, der AWS CLI, oder der ElastiCache API erwerben können.

Wichtig

Wenn Sie den Beispielen in diesem Abschnitt folgen, fallen Gebühren auf Ihrem AWS Konto an, die Sie nicht rückgängig machen können.

Themen

- [Kauf eines reservierten Knotens \(Konsole\)](#)
- [Kauf eines reservierten Knotens \(AWS CLI\)](#)
- [Kauf eines reservierten Knotens \(ElastiCache API\)](#)

Kauf eines reservierten Knotens (Konsole)

Das folgende Beispiel zeigt den Kauf eines bestimmten reservierten Knotens, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der reservierten Knoten-ID myreservationID.

Das folgende Verfahren verwendet die AWS Management Console, um das Angebot für reservierte Knoten zu erwerben, indem die ID angeboten wird.

So kaufen Sie reservierte Knoten

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der Navigationsliste den Link Reserved Nodes (Reservierte Knoten) aus.
3. Wählen Sie die Schaltfläche Purchase Reserved Nodes (Reservierte Knoten kaufen) aus.
4. Wählen Sie für Engine Valkey, Redis oder OSS Memcached aus.
5. Treffen Sie zur Ermittlung der verfügbaren Angebote eine Auswahl für die folgenden Optionen:
 - Node Type
 - Laufzeit
 - Angebotstyp
 - Eine optionale Reserved-Node-ID

Nachdem Sie Ihre Auswahl getroffen haben, werden die Kosten pro Knoten und die Gesamtknoten der ausgewählten Optionen unter Reservation details (Reservierungsdetails) angezeigt.

6. Klicken Sie auf Purchase (Kaufen).

Kauf eines reservierten Knotens (AWS CLI)

Das folgende Beispiel zeigt den Erwerb des bestimmten reservierten Clusters, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der Knoten-ID myreservationID.

Geben Sie den folgenden Befehl an einer Eingabeaufforderung ein:

Für Linux, macOS oder Unix:

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

Für Windows:

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

Daraufhin erhalten Sie ein Ergebnis, das dem hier dargestellten entspricht:

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

Weitere Informationen finden Sie unter [purchase-reserved-cache-nodes-offering](#) in der AWS CLI Referenz.

Kauf eines reservierten Knotens (ElastiCache API)

Das folgende Beispiel zeigt den Kauf des bestimmten reservierten Knotens, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der reservierten Cluster-ID myreservationID.

Rufen Sie die `PurchaseReservedCacheNodesOffering`-Operation mit folgenden Parametern auf:

- `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>
```

Weitere Informationen finden Sie [PurchaseReservedCacheNodesOffering](#) in der ElastiCache API Referenz.

Abrufen von Informationen über Ihre reservierten Knoten

Informationen zu den reservierten Knoten, die Sie gekauft haben AWS Management Console, erhalten Sie über die AWS CLI, und die ElastiCache API.

Themen

- [Abrufen von Informationen über Ihre reservierten Knoten \(Konsole\)](#)
- [Abrufen von Informationen über Ihre reservierten Knoten \(AWS CLI\)](#)
- [Informationen über Ihre reservierten Knoten abrufen \(ElastiCache API\)](#)

Abrufen von Informationen über Ihre reservierten Knoten (Konsole)

Das folgende Verfahren beschreibt, wie Sie mit dem Informationen AWS Management Console zu den von Ihnen gekauften reservierten Knoten abrufen können.

So rufen Sie Informationen über Ihre gekauften reservierten Knoten ab

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der Navigationsliste den Link Reserved Nodes (Reservierte Knoten) aus.

Die reservierten Knoten für Ihr Konto werden in der Liste "Reserved Nodes" (Reservierte Knoten) aufgeführt. Sie können einen beliebigen reservierten Knoten in der Liste auswählen, um sich detaillierte Informationen zu diesem Knoten im Detailbereich am unteren Rand der Konsole anzeigen zu lassen.

Abrufen von Informationen über Ihre reservierten Knoten (AWS CLI)

Um Informationen über reservierte Knoten für Ihr AWS Konto zu erhalten, geben Sie an der Befehlszeile den folgenden Befehl ein:

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

Dieser Vorgang erzeugt eine Ausgabe, die der folgenden ähnelt (JSONFormat):

```
{
  "ReservedCacheNodeId": "myreservationid",
  "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
```

```
"CacheNodeType": "cache.xx.small",
"DataTiering": "disabled",
"Duration": "31536000",
"ProductDescription": "memcached",
"OfferingType": "Medium Utilization",
"MaxRecords": 0
}
```

Weitere Informationen finden Sie unter [describe-- reserved-cache-nodes](#) in der AWS CLI Referenz.

Informationen über Ihre reservierten Knoten abrufen (ElastiCache API)

Rufen Sie den DescribeReservedCacheNodes Vorgang auf, um Informationen über reservierte Knoten für Ihr AWS Konto zu erhalten.

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>
```

Weitere Informationen finden Sie [DescribeReservedCacheNodes](#) in der ElastiCache API Referenz.

Migration von Knoten der vorherigen Generation

Knoten der vorherigen Generation sind Knotentypen, die nicht mehr verwendet werden. Wenn Sie keine vorhandenen Cluster haben, die einen Knotentyp der vorherigen Generation verwenden, ElastiCache unterstützt die Erstellung neuer Cluster mit diesem Knotentyp nicht.

Aufgrund der begrenzten Anzahl von Node-Typen der vorherigen Generation können wir keinen erfolgreichen Ersatz garantieren, wenn ein Node in Ihrem(n) Cluster(n) fehlerhaft wird. In einem solchen Szenario kann die Verfügbarkeit Ihres Clusters negativ beeinflusst werden.

Wir empfehlen, dass Sie Ihre(n) Cluster auf einen neuen Knotentyp migrieren, um die Verfügbarkeit und Leistung zu verbessern. Einen empfohlenen Knotentyp für die Migration finden Sie unter [Upgrade-Pfade](#). Eine vollständige Liste der unterstützten Knotentypen und Knotentypen der vorherigen Generation finden Sie unter [Unterstützte Knotentypen](#). ElastiCache

Migrieren von Knoten auf einem Valkey- oder Redis-Cluster OSS

Das folgende Verfahren beschreibt, wie Sie Ihren Valkey- oder OSS Redis-Cluster-Knotentyp mithilfe der Konsole migrieren. ElastiCache Während dieses Vorgangs bearbeitet Ihr Valkey- oder OSS Redis-Cluster weiterhin Anfragen mit minimaler Ausfallzeit. Abhängig von Ihrer Clusterkonfiguration können folgende Ausfallzeiten auftreten. Die folgenden Angaben sind Schätzungen und können je nach Ihrer spezifischen Konfiguration abweichen:

- Bei deaktiviertem Clustermodus (einzelner Knoten) kann es zu etwa 60 Sekunden dauern, was hauptsächlich auf die Ausbreitung zurückzuführen DNS ist.
- Bei deaktiviertem Clustermodus (mit Replikatknoten) kann es bei Clustern, auf denen Valkey 7.2 und höher oder Redis OSS 5.0.6 und höher ausgeführt wird, etwa 1 Sekunde dauern. Bei allen älteren Versionen kann es etwa 10 Sekunden dauern.
- Bei aktiviertem Cluster-Modus kann etwa 1 Sekunde angezeigt werden.

So ändern Sie einen Valkey- oder OSS Redis-Cluster-Knotentyp mithilfe der Konsole:

1. Melden Sie sich bei der Konsole an und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.
3. Wählen Sie aus der Cluster-Liste den gewünschten Cluster für die Migration aus.
4. Wählen Sie Aktionen und dann Ändern.
5. Wählen Sie den neuen Knotentyp aus der Knotentypliste aus.
6. Wenn Sie den Migrationsprozess sofort durchführen möchten, wählen Sie Sofort anwenden. Wenn die Option Apply immediately (Sofort anwenden) nicht gewählt wird, wird der Migrationsprozess während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus. Wenn Sie im vorherigen Schritt Apply immediately aktiviert haben, ändert sich der Cluster-Status zu modifying. Wenn der Status zu available wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

Um einen Valkey- oder OSS Redis-Cluster-Knotentyp zu ändern, verwenden Sie: AWS CLI

Verwenden Sie [modify-replication-group](#) API wie folgt:

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group /  
  --replication-group-id my-replication-group /  
  --cache-node-type new-node-type /  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^  
  --replication-group-id my-replication-group ^  
  --cache-node-type new-node-type ^  
  --apply-immediately
```

In diesem Szenario ist der Wert von *new-node-type* der Knotentyp, zu dem Sie migrieren. Durch die Übergabe des `--apply-immediately`-Parameters wird die Aktualisierung sofort angewendet, wenn die Replikationsgruppe vom Status geändert in den Status verfügbar wechselt. Wenn die Option `Apply immediately` (Sofort anwenden) nicht gewählt wird, wird der Migrationsprozess während des nächsten Wartungsfensters des Clusters durchgeführt.

Note

Wenn Sie den Cluster nicht ändern können, weil ein `InvalidCacheClusterState`-Fehler aufgetreten ist, müssen Sie zuerst einen fehlerhaften Wiederherstellungsknoten entfernen.

Reparieren oder Entfernen von `restore-failed-node` (en)

Das folgende Verfahren beschreibt, wie Sie Knoten, die bei der Wiederherstellung ausgefallen sind, reparieren oder aus Ihrem Valkey- oder Redis-Cluster entfernen. OSS Weitere Informationen darüber, wie ElastiCache Knoten in den Status „Wiederherstellung fehlgeschlagen“ übergehen, finden Sie unter [ElastiCache Knotenstatus anzeigen](#). Wir empfehlen, zuerst alle Knoten zu entfernen, die sich in einem Zustand befinden, in dem eine Wiederherstellung fehlgeschlagen ist, dann die verbleibenden Knoten der vorherigen Generation im ElastiCache Cluster auf einen Knotentyp der neueren Generation zu migrieren und schließlich die erforderliche Anzahl von Knoten wieder hinzuzufügen.

So entfernen Sie den Wiederherstellungsfehler-Knoten (Konsole):

1. Melden Sie sich bei der Konsole an und öffnen Sie die Konsole unter ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.
3. Wählen Sie in der Cluster-Liste den Cluster aus, aus dem Sie einen Knoten entfernen möchten.
4. Wählen Sie aus der Shard-Liste den Shard aus, aus dem Sie einen Knoten entfernen möchten. Überspringen Sie diesen Schritt, wenn der Cluster-Modus für den Cluster deaktiviert ist.
5. Wählen Sie aus der Knotenliste den Knoten mit einem Status `restore-failed` aus.
6. Wählen Sie Aktionen und wählen Sie dann Knoten löschen.

Nachdem Sie die Knoten, bei denen die Wiederherstellung fehlgeschlagen ist, aus Ihrem ElastiCache Cluster entfernt haben, können Sie jetzt zu einem neueren Generationstyp migrieren. Weitere Informationen finden Sie weiter oben unter [Migrieren von Knoten auf einem Valkey- oder Redis-Cluster OSS](#).

Informationen zum Hinzufügen von Knoten zu Ihrem ElastiCache Cluster finden Sie unter [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)

Migrieren von Knoten auf einen Memcached-Cluster

Um ElastiCache (Memcached) zu einem anderen Knotentyp zu migrieren, müssen Sie einen neuen Cluster erstellen, der immer leer ist und von Ihrer Anwendung aufgefüllt werden kann.

So migrieren Sie Ihren ElastiCache (Memcached) Cluster-Knotentyp mithilfe der Konsole:
ElastiCache

- Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).
- Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#).
- Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Löschen eines Clusters in ElastiCache](#)

Verwaltung von Clustern in ElastiCache

Ein Cluster ist eine Sammlung von einem oder mehreren Cache-Knoten, auf denen alle eine Instanz der Valkey-, Redis OSS - oder Memcached Engine-Software ausführen. Wenn Sie einen Cluster erstellen, geben Sie die Engine und die Version an, die von allen Knoten verwendet werden soll.

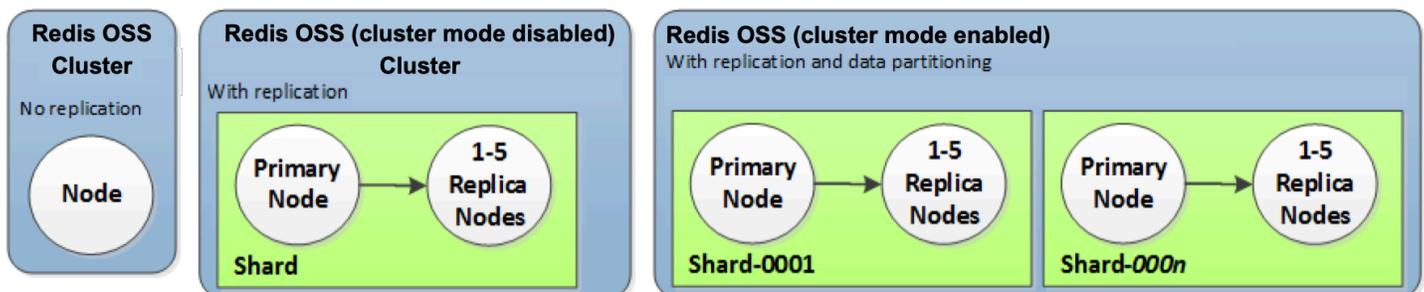
Valkey- und Redis-Cluster OSS

Das folgende Diagramm zeigt einen typischen Valkey- oder Redis-Cluster. OSS Diese Cluster können einen einzelnen Knoten oder bis zu sechs Knoten innerhalb eines Shards (API/CLI: Knotengruppe) enthalten. Ein Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem Knoten hat keinen Shard, und ein Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit mehreren Knoten hat einen einzelnen Shard. Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) können bis zu 500 Shards enthalten, wobei Ihre Daten auf die Shards aufgeteilt sind. Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Engine-Version Valkey 7.2 und höher oder Redis 5.0.6 und höher ist. OSS Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikat pro Shard) und 500 Shards (ein primärer Knoten und keine Replikat) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#). Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

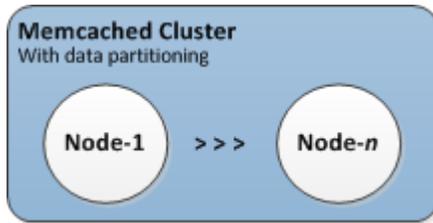
Wenn Sie mehrere Knoten in einem Valkey- oder OSS Redis-Shard haben, ist einer der Knoten ein primärer Knoten mit Lese-/Schreibzugriff. Alle anderen Knoten im Shard sind schreibgeschützte Replikat.

Typische Valkey- oder Redis-Cluster sehen wie folgt aus. OSS



Memcached-Cluster

Typische Memcached-Cluster sehen wie folgt aus. Memcached-Cluster enthalten 1 bis 60 Knoten, über die Sie Ihre Daten horizontal partitionieren.



Elasticache-Operationen für Valkey, Redis und Memcached OSS

Die meisten ElastiCache Operationen werden auf Clusterebene ausgeführt. Sie können einen Cluster mit einer bestimmten Anzahl von Knoten und einer Parametergruppe einrichten, die die Eigenschaften für jeden Knoten steuert. Alle Knoten innerhalb eines Clusters gehören demselben Knotentyp an und verfügen über die gleichen Einstellungen für Parameter und Sicherheitsgruppen.

Jeder Cluster muss über eine Cluster-Kennung verfügen. Die Cluster-Kennung ist ein vom Kunden angegebener Name für den Cluster. Diese Kennung gibt bei der Interaktion mit den AWS CLI Befehlen ElastiCache API und einen bestimmten Cluster an. Die Cluster-ID muss für diesen Kunden in einer AWS Region eindeutig sein.

ElastiCache unterstützt mehrere Engine-Versionen. Sofern es keine bestimmten Gründe dagegen gibt, raten wir dazu, die neueste Version zu verwenden.

ElastiCache Cluster sind für den Zugriff über eine EC2 Amazon-Instance konzipiert. Wenn Sie Ihren Cluster in einer virtuellen privaten Cloud (VPC) starten, die auf dem VPC Amazon-Service basiert, können Sie von außen darauf zugreifen AWS. Weitere Informationen finden Sie unter [Zugriff auf ElastiCache Ressourcen von außen AWS](#).

Eine Liste der unterstützten Versionen finden Sie unter [Unterstützte Engines und Versionen](#), [Unterstützte Redis-Versionen OSS](#), und [Unterstützte ElastiCache \(Memcached\) Versionen](#).

Auswahl eines Netzwerktyps in ElastiCache

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er Folgendes akzeptiert:

- nur IPv4 Verbindungen,
- nur IPv6 Verbindungen,
- IPv4 sowohl als auch IPv6 Verbindungen (Dual-Stack)

[IPv6 wird für Workloads unterstützt, die Valkey 7.2 und höher oder Redis OSS Engine Version 6.2 und höher auf allen auf dem Nitro-System basierenden Instanzen verwenden.](#) Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6

Note

Die Migration von Clustern, die vor der Verfügbarkeit von IPv6/Dual-Stack erstellt wurden, wird nicht unterstützt. Das Umschalten zwischen Netzwerktypen in neu erstellten Clustern wird ebenfalls nicht unterstützt.

[IPv6 wird für Workloads unterstützt, die die Memcached Engine ab Version 1.6.6 auf allen auf dem Nitro-System basierenden Instanzen verwenden.](#) Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6

Konfigurieren von Subnetzen für den Netzwerktyp

Wenn Sie einen Cluster in einem Amazon erstellen VPC, müssen Sie eine Subnetzgruppe angeben. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only. IPv6

Verwenden von Dual-Stack

Wenn Sie ElastiCache (RedisOSS) im aktivierten Clustermodus verwenden, unterscheidet sich die Verbindung mit allen Clusterknoten über den Konfigurationsendpunkt aus Sicht einer Anwendung nicht von der direkten Verbindung zu einem einzelnen Cache-Knoten. Um dies zu

erreichen, muss ein clusterfähiger Client einen Cluster-Erkennungsprozess durchführen und die Konfigurationsinformationen für alle Knoten anfordern. Das Erkennungsprotokoll von Redis unterstützt nur eine IP pro Knoten.

Wenn Sie einen Cache-Cluster mit ElastiCache (Memcached) erstellen und Dual-Stack als Netzwerktyp wählen, müssen Sie anschließend einen IP-Erkennungstyp festlegen — entweder IPv4 oder IPv6. ElastiCache verwendet standardmäßig den Netzwerktyp und die IP-Erkennung IPv6, aber das kann geändert werden. Wenn Sie Auto Discovery verwenden, werden nur die IP-Adressen des von Ihnen gewählten IP-Typs an den Memcached-Client zurückgegeben. Weitere Informationen finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#).

Um die Abwärtskompatibilität mit allen vorhandenen Clients aufrechtzuerhalten, wurde die IP-Erkennung eingeführt, sodass Sie den IP-Typ (d. h. IPv4 oder IPv6) auswählen können, der im Discovery-Protokoll angekündigt werden soll. Dadurch wird die auto Erkennung zwar auf nur einen IP-Typ beschränkt, aber Dual-Stack ist für Workloads mit aktiviertem Clustermodus dennoch von Vorteil, da es Migrationen (oder Rollbacks) von einem IPv4 zu einem IPv6 Discovery-IP-Typ ohne Ausfallzeiten ermöglicht.

TLSaktivierte Dual-Stack-Cluster ElastiCache

Wenn für ElastiCache Cluster aktiviert TLS ist, geben Cluster-Erkennungsfunktionen wie `cluster slots`, `cluster shards`, und `cluster nodes` mit Valkey oder Redis OSS und `config get cluster` mit Memcached Hostnamen statt IPs. Die Hostnamen werden dann verwendet, anstatt eine Verbindung zu den IPs zum Cluster herzustellen und einen Handshake durchzuführen. ElastiCache TLS bedeutet, dass Clients nicht vom IP-Discovery-Parameter betroffen sind. Bei TLS aktivierten Clustern hat der IP Discovery-Parameter keine Auswirkung auf das bevorzugte IP-Protokoll. Stattdessen wird das verwendete IP-Protokoll davon bestimmt, welches IP-Protokoll der Client bei der Auflösung von DNS Hostnamen bevorzugt.

Beispiele zur Konfiguration einer IP-Protokollpräferenz bei der Auflösung von DNS Hostnamen finden Sie unter [TLSaktivierte ElastiCache Dual-Stack-Cluster](#)

Verwenden von AWS Management Console (Valkey und Redis) OSS

Wenn Sie einen Cluster mit dem erstellen AWS Management Console, wählen Sie unter Konnektivität entweder IPv4 als Netzwerktyp oder IPv6 oder Dual-Stack. Wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) erstellen und Dual-Stack wählen, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv4 oder IPv6.

Weitere Informationen finden Sie unter [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#) oder [Einen Valkey oder Redis erstellen OSS \(Clustermodus deaktiviert\) \(Konsole\)](#).

Wählen Sie beim Erstellen einer Replikationsgruppe mit dem AWS Management Console einen Netzwerktyp IPv4, entweder IPv6 oder Dual-Stack. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Weitere Informationen finden Sie unter [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) von Grund auf neu erstellen](#) oder [Eine Replikationsgruppe in Valkey oder Redis OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen](#).

Verwenden von AWS Management Console (Memcached)

Wenn Sie einen Cache-Cluster mit dem erstellen AWS Management Console, wählen Sie unter Konnektivität entweder IPv4 einen Netzwerktyp IPv6 oder Dual-Stack. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

Verwenden Sie das CLI mit Valkey, Redis OSS oder Memcached

Redis OSS

Wenn Sie einen Cache-Cluster mit Valkey oder Redis OSS mithilfe von erstellen CLI, verwenden Sie den [create-cache-cluster](#) Befehl und geben die NetworkType Parameter und an: IPDiscovery

Für Linux, macOS oder 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
```

Für 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
```

Wenn Sie mit dem eine Replikationsgruppe mit deaktiviertem Clustermodus erstellenCLI, verwenden Sie den [create-replication-group](#)Befehl und geben die Parameter NetworkType und IPDiscovery an:

Für Linux, macOS oder 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
```

Für 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
```

Wenn Sie eine Replikationsgruppe mit aktiviertem Clustermodus erstellen und IPv4 für die IP-Erkennung mit dem verwendenCLI, verwenden Sie den [create-replication-group](#)Befehl und geben die IPDiscovery Parameter NetworkType und an:

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \
  --replication-group-id demo-cluster \
  --replication-group-description "demo cluster" \
```

```
--cache-node-type cache.m5.large \  
--num-node-groups 2 \  
--engine redis \  
--cache-subnet-group-name xyz \  
--network-type dual_stack \  
--ip-discovery ipv4 \  
--region us-east-1
```

Für 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
```

Wenn Sie eine Replikationsgruppe mit aktiviertem Clustermodus erstellen und IPv6 für die IP-Erkennung mit dem verwendenCLI, verwenden Sie den [create-replication-group](#) Befehl und geben die IPDiscovery Parameter NetworkType und an:

Für Linux, macOS oder 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
```

Für 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

Wenn Sie mit Memcached einen Cache-Cluster mit dem erstellenCLI, verwenden Sie den [create-cache-cluster](#)Befehl und geben die Parameter und an: NetworkType IPDiscovery

Für Linux, macOS oder 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
```

Für 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
```

Identifizieren Sie automatisch Knoten in Ihrem Cluster (Memcached)

ElastiCache Unterstützt bei Clustern, auf denen die Memcached-Engine ausgeführt wird, Auto Discovery — die Fähigkeit von Client-Programmen, automatisch alle Knoten in einem Cache-Cluster zu identifizieren und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten.

Note

Auto Discovery wurde für Cache-Cluster hinzugefügt, die auf Amazon ElastiCache Memcached ausgeführt werden. Auto Discovery ist für Valkey- oder Redis-Engines nicht verfügbar. OSS

Mit Auto Discovery muss Ihre Anwendung keine manuelle Verbindung mit einzelnen Cache-Knoten herstellen. Stattdessen stellt sie eine Verbindung mit einem Memcached-Knoten her und ruft die Liste der Knoten ab. Anhand dieser Liste erhält Ihre Anwendung Informationen über die restlichen Knoten im Cluster und kann eine Verbindung mit einem beliebigen Knoten herstellen. Die einzelnen Endpunkte der Cache-Knoten in Ihrer Anwendung müssen nicht als vordefinierter Code aufgenommen werden.

Wenn Sie in Ihrem Cluster einen Dual-Stack-Netzwerktyp verwenden, gibt Auto Discovery nur IPv4 oder IPv6 Adressen zurück, je nachdem, welche Sie auswählen. Weitere Informationen finden Sie unter [Auswahl eines Netzwerktyps in ElastiCache](#).

Alle Cache-Knoten im Cluster führen eine Liste mit Metadaten über alle anderen Knoten. Diese Metadaten werden aktualisiert, sobald Knoten hinzugefügt oder aus dem Cluster entfernt werden.

Themen

- [Vorteile von Auto Discovery mit Memcached](#)
- [Funktionsweise von Auto Discovery](#)
- [Verwenden von Auto Discovery](#)
- [Manuelles Herstellen einer Verbindung zu Memached-Cache-Knoten](#)
- [Hinzufügen von Auto Discovery zu Ihrer Memcached-Clientbibliothek](#)
- [ElastiCache Kunden mit auto Erkennung](#)

Vorteile von Auto Discovery mit Memcached

Bei der Verwendung von Memcached bietet Auto Discovery die folgenden Vorteile:

- Wenn Sie die Zahl der Knoten in einem Cache-Cluster erhöhen, registrieren sich die neuen Knoten mit dem Konfigurationsendpunkt und mit allen anderen Knoten selbst. Wenn Sie Knoten aus dem Cache-Cluster entfernen, heben die entsprechenden Cluster ihre Registrierung selbst auf. In beiden Fällen werden alle anderen Knoten im Cluster mit den neuen Cache-Knotenmetadaten aktualisiert.
- Ausfälle von Cache-Knoten werden automatisch erkannt und ausgefallene Knoten automatisch ersetzt.

Note

Der Knoten fällt solange aus, bis er ersetzt wird.

- Ein Client-Programm muss nur eine Verbindung mit dem Konfigurationsendpunkt herstellen. Anschließend stellt die Auto Discovery-Bibliothek eine Verbindung mit allen anderen Knoten im Cluster her.
- Client-Programme fragen den Cluster einmal pro Minute ab (dieses Intervall kann bei Bedarf geändert werden). Wenn Änderungen an der Cluster-Konfiguration vorgenommen wurden, z. B. neue oder gelöschte Knoten, erhält der Client eine aktualisierte Liste mit Metadaten. Anschließend stellt der Client je nach Bedarf eine Verbindung mit diesen Knoten her bzw. trennt die Verbindung.

Auto Discovery ist auf allen ElastiCache Memcached-Cache-Clustern aktiviert. Sie müssen Ihre Cache-Knoten nicht neu starten, um diese Funktion zu verwenden.

Funktionsweise von Auto Discovery

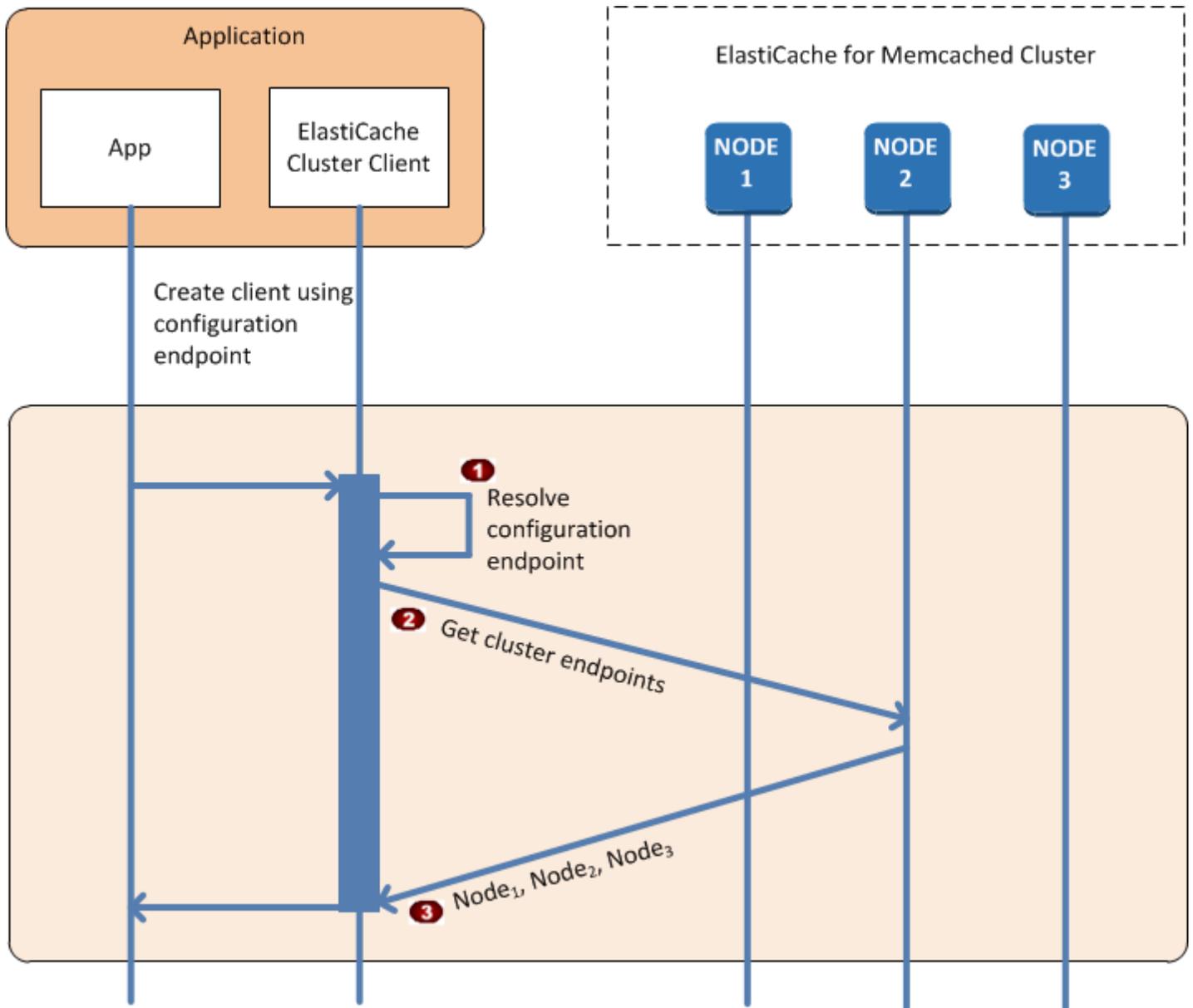
Themen

- [Herstellen von Verbindungen mit Cache-Knoten](#)
- [Normale Cluster-Operationen](#)
- [Weitere Operationen in](#)

In diesem Abschnitt wird beschrieben, wie Clientanwendungen den ElastiCache Clusterclient verwenden, um Cache-Knotenverbindungen zu verwalten und mit Datenelementen im Cache zu interagieren.

Herstellen von Verbindungen mit Cache-Knoten

Von der Anwendungsseite aus betrachtet, unterscheidet sich das Herstellen einer Verbindung mit dem Cluster-Konfigurationsendpunkt nicht vom Herstellen einer direkten Verbindung mit einem einzelnen Cache-Knoten. Das folgende Ablaufdiagramm zeigt den Prozess zum Herstellen der Verbindung mit Cache-Knoten.



Prozess zum Herstellen von Verbindungen mit Cache-Knoten

- Die Anwendung löst den DNS Namen des Konfigurationsendpunkts auf. Da der Konfigurationsendpunkt CNAME Einträge für alle Cache-Knoten verwaltet, wird der DNS Name in einen der Knoten aufgelöst. Der Client kann dann eine Verbindung zu diesem Knoten herstellen.
- Der Client fordert die Konfigurationsinformationen für alle anderen Knoten an. Da jeder Knoten Konfigurationsinformationen für alle Knoten im Cluster verwaltet, kann jeder Knoten auf Anfrage Konfigurationsinformationen an den Client weitergeben.

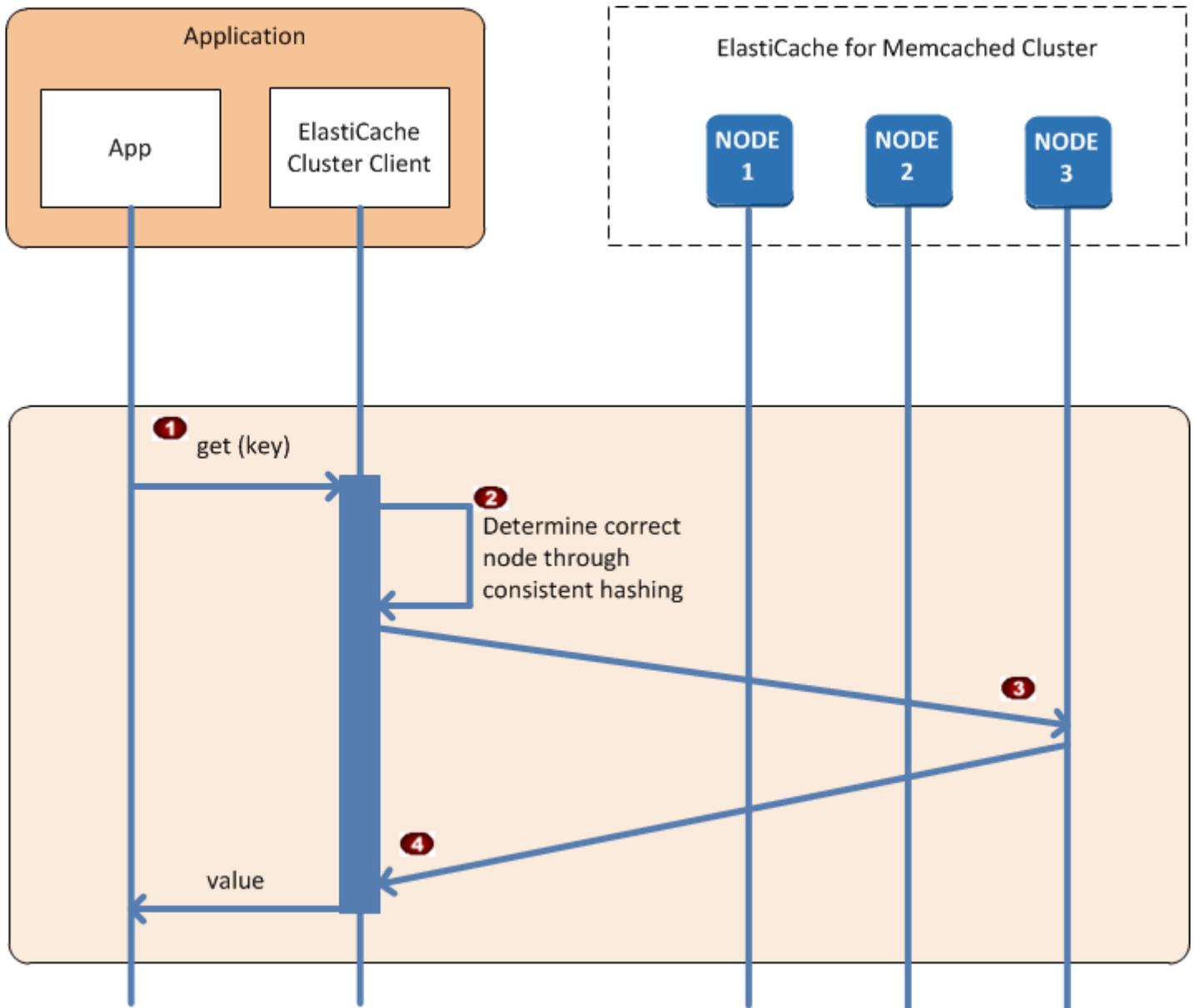
- Der Client empfängt die aktuelle Liste mit den Hostnamen und IP-Adressen der Cache-Knoten. Anschließend kann er eine Verbindung mit allen anderen Knoten im Cluster herstellen.

Note

Das Client-Programm aktualisiert seine Liste mit den Hostnamen und IP-Adressen der Cache-Knoten einmal pro Minute. Dieses Abfrageintervall kann bei Bedarf angepasst werden.

Normale Cluster-Operationen

Wenn die Anwendung eine Verbindung zu allen Cache-Knoten hergestellt hat, bestimmt der ElastiCache Cluster-Client, auf welchen Knoten einzelne Datenelemente gespeichert werden sollen und welche Knoten später nach diesen Datenelementen abgefragt werden sollen. Das folgende Ablaufdiagramm zeigt den Prozess für normale Cluster-Operationen.



Prozess für normale Cluster-Operationen

- Die Anwendung erstellt eine get-Anforderung für ein bestimmtes Datenelement, das durch seinen Schlüssel identifiziert wird.
- Der Client verwendet einen Hashing-Algorithmus für den Schlüssel, um den Cache-Knoten zu ermitteln, der das Datenelement enthält.
- Das Datenelement wird vom entsprechenden Knoten angefordert.
- Das Datenelement wird an die Anwendung zurückgegeben.

Weitere Operationen in

In manchen Situationen kann es vorkommen, dass Sie eine Änderung an den Knoten eines Clusters vornehmen. So können Sie beispielsweise einen zusätzlichen Knoten hinzufügen, um zusätzlichen Bedarf zu decken, oder einen Knoten löschen, um in Zeiten geringerer Nachfrage Geld zu sparen. Oder Sie ersetzen einen Knoten, weil er auf die eine oder andere Weise ausgefallen ist.

Bei einer Änderung im Cluster, die eine Aktualisierung der Metadaten auf die Endpunkte des Clusters erfordert, wird diese Änderung auf allen Knoten gleichzeitig vorgenommen. So sind die Metadaten in einem bestimmten Knoten konsistent mit den Metadaten in allen anderen Knoten im Cluster.

In jedem Fall sind die Metadaten für alle Knoten jederzeit konsistent, da sie für alle Knoten im Cluster gleichzeitig aktualisiert werden. Verwenden Sie immer den Konfigurationsendpunkt, um die Endpunkte der verschiedenen Knoten im Cluster zu erhalten. Durch Verwenden des Konfigurationsendpunkts stellen Sie sicher, dass Sie keine Endpunktdaten von einem Knoten erhalten, der "verschwindet".

Hinzufügen eines Knotens

Während der Zeit, in der der Knoten eingerichtet wird, ist sein Endpunkt nicht in den Metadaten enthalten. Sobald der Knoten verfügbar ist, wird er den jeweiligen Metadaten der Cluster-Knoten hinzugefügt. In diesem Szenario sind die Metadaten für alle Knoten konsistent und Sie können mit dem neuen Knoten erst interagieren, sobald er verfügbar ist. Vorher liegen Ihnen keine Informationen darüber vor und Sie interagieren mit den Knoten in Ihrem Cluster so, als ob der neue Knoten nicht vorhanden wäre.

Löschen eines Knotens

Wenn ein Knoten entfernt wird, wird sein Endpunkt erst in den Metadaten gelöscht und anschließend wird der Knoten aus dem Cluster entfernt. In diesem Szenario sind die Metadaten in allen Knoten konsistent und zu keiner Zeit enthalten sie den Endpunkt für den Knoten, der entfernt werden soll, während der Knoten nicht verfügbar ist. Während des Zeitraums, in dem der Knoten entfernt wird, erfolgt keine Meldung in den Metadaten. Demzufolge interagiert Ihre Anwendung nur mit den verbleibenden $n-1$ Knoten, so als ob der Knoten nicht vorhanden wäre.

Ersetzen eines Knotens

Wenn ein Knoten ausfällt, wird ElastiCache dieser Knoten heruntergefahren und ein Ersatzknoten installiert. Das Ersetzen dauert einige Minuten. Während dieser Zeit zeigen die Metadaten in allen

Knoten den Endpunkt für den ausgefallenen Knoten noch an, doch jeder Versuch, mit dem Knoten zu interagieren, schlägt fehl. Daher sollte Ihre Logik immer Logik für Wiederholungsversuche umfassen.

Verwenden von Auto Discovery

Gehen Sie wie folgt vor, um Auto Discovery mit ElastiCache (Memcached) zu verwenden:

- [Rufen Sie den Konfigurationsendpunkt ab](#)
- [Laden Sie den ElastiCache Cluster-Client herunter](#)
- [Ändern Sie Ihr Anwendungsprogramm](#)

Rufen Sie den Konfigurationsendpunkt ab

Zum Herstellen der Verbindung mit einem Cluster müssen die Client-Programme den Cluster-Konfigurationsendpunkt kennen. Weitere Informationen finden Sie im Thema [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#).

Sie können auch den Befehl `aws elasticache describe-cache-clusters` mit dem Parameter `--show-cache-node-info` verwenden:

Unabhängig von der Methode, mit der Sie die Endpunkte des Clusters suchen, enthält der Konfigurationsendpunkt in seiner Adresse immer `.cfg`.

Example Finden von Endpunkten mit dem for AWS CLI ElastiCache

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

Dieser Vorgang erzeugt eine Ausgabe, die der folgenden ähnelt (JSONFormat):

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

Laden Sie den ElastiCache Cluster-Client herunter

Um Auto Discovery nutzen zu können, müssen Client-Programme den ElastiCacheClusterclient verwenden. Der ElastiCache Clusterclient ist für JavaPHP, und verfügbar. NETund enthält die gesamte notwendige Logik, um all Ihre Cache-Knoten zu finden und eine Verbindung zu ihnen herzustellen.

Um den ElastiCache Cluster-Client herunterzuladen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der ElastiCache Konsole ElastiCache Cluster Client und dann Herunterladen aus.

Der Quellcode für den ElastiCache Cluster-Client für Java ist unter <https://github.com/amazonwebservices/aws-elasticache-cluster-client-> verfügbarmemcached-for-java. Diese Bibliothek basiert auf dem häufig verwendeten Spymemcached-Client. Der ElastiCache Cluster-Client ist unter der Amazon-Softwarelizenz <https://aws.amazon.com/asl> veröffentlicht. Es steht Ihnen frei, den Quellcode nach eigenen Wünschen zu ändern. Sie können den Code sogar in andere Memcached-Open-Source-Bibliotheken oder in Ihren eigenen Client-Code integrieren.

Note

Um den ElastiCache Cluster Client für verwenden zu könnenPHP, müssen Sie ihn zunächst auf Ihrer EC2 Amazon-Instance installieren. Weitere Informationen finden Sie unter [Installieren des ElastiCache-Cluster-Clients-für-PHP](#).

Laden Sie für einen TLS unterstützten Client die Binärdatei mit PHP Version 7.4 oder höher herunter.

Um den ElastiCache Cluster-Client für zu verwenden. NET, müssen Sie es zuerst auf Ihrer EC2 Amazon-Instance installieren. Weitere Informationen finden Sie unter [Installation des ElastiCache Clusterclients für. NET](#).

Ändern Sie Ihr Anwendungsprogramm

Ändern Sie Ihr Anwendungsprogramm, sodass es Auto Discovery verwendet. In den folgenden Abschnitten wird die Verwendung des ElastiCache Cluster-Clients für JavaPHP, und beschrieben. NET.

Important

Stellen Sie beim Angeben des Cluster-Konfigurationsendpunkts sicher, dass der Endpunkt in seiner Adresse ".cfg" enthält, wie hier dargestellt. Verwenden Sie keinen Endpunkt CNAME oder keinen Endpunkt ohne „.cfg“.

```
"mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
```

Wenn Sie den Cluster-Konfigurationsendpunkt nicht explizit angeben, erfolgt die Konfiguration für einen spezifischen Knoten.

Den ElastiCache Cluster-Client für Java verwenden

Das folgende Programm zeigt, wie Sie den ElastiCache Cluster-Client verwenden, um eine Verbindung zu einem Cluster-Konfigurationsendpunkt herzustellen und dem Cache ein Datenelement hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

```
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");
}
}

```

Verwenden des ElastiCache Cluster-Clients für PHP

Das folgende Programm zeigt, wie Sie den ElastiCache Cluster-Client verwenden, um eine Verbindung zu einem Clusterkonfigurationsendpunkt herzustellen und dem Cache ein Datenelement hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

Um den ElastiCache Cluster Client für verwenden zu können PHP, müssen Sie ihn zunächst auf Ihrer EC2 Amazon-Instance installieren. Weitere Informationen finden Sie unter [Installieren des ElastiCache-Cluster-Clients-für-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');
?>
```

Ein Beispiel für die Verwendung des ElastiCache Cluster-Clients bei TLS aktivierter Option finden Sie unter [In-Transit-Verschlüsselung mit PHP und Memcached verwenden](#).

Verwenden des ElastiCache Cluster-Clients für .NET

Note

Die ElastiCache .NET Der Cluster-Client ist seit Mai 2022 veraltet.

.NET Der Client für ElastiCache ist Open Source unter <https://github.com/awslabs/elasticache-cluster-config-net>.

.NET Anwendungen beziehen ihre Konfigurationen normalerweise aus ihrer Konfigurationsdatei. Nachfolgend finden Sie ein Beispiel einer Anwendungs-Config-Datei.

```
<?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>
```

Das unten stehende C#-Programm zeigt, wie Sie den ElastiCache Cluster-Client verwenden, um eine Verbindung zu einem Cluster-Konfigurationsendpunkt herzustellen und dem Cache ein Datenelement hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

```
// *****
// Sample C# code to show how to integrate with the Amazon ElastiCache 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
```

Manuelles Herstellen einer Verbindung zu Memcached-Cache-Knoten

Wenn Ihr Client-Programm Auto Discovery nicht verwendet, kann es manuell eine Verbindung zu jedem der Memcache-Cache-Knoten herstellen. Dies ist das Standardverhalten für Memcached-Clients.

Eine Liste der Hostnamen und Portnummern der Cache-Knoten erhalten Sie über die [AWS Managementkonsole](#). Sie können den AWS CLI `aws elasticache describe-cache-clusters` Befehl auch mit dem `--show-cache-node-info` Parameter verwenden.

Example

Der folgende Java-Codeausschnitt zeigt, wie eine Verbindung mit allen Knoten in einem aus vier Knoten bestehenden Cache-Cluster hergestellt wird:

```
...  
  
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

Wenn Sie Ihren Cache-Cluster durch Hinzufügen oder Entfernen von Knoten nach oben oder nach unten skalieren, müssen Sie die Liste der Knoten im Client-Code aktualisieren.

Hinzufügen von Auto Discovery zu Ihrer Memcached-Clientbibliothek

Die Konfigurationsinformationen für Auto Discovery werden redundant in jedem Memcached-Cache-Clusterknoten gespeichert. Client-Anwendungen können jeden Cache-Knoten abfragen und die Konfigurationsinformationen für alle Knoten im Cluster abrufen.

Die jeweilige Methode, die eine Anwendung dazu nutzt, hängt von der Cache-Engine-Version ab:

- Wenn die Cache-Engine-Version 1.4.14 oder höher ist, verwenden Sie den Befehl `config`.
- Wenn die Cache-Engine-Version älter als 1.4.14 ist, verwenden Sie den Befehl `get AmazonElastiCache:cluster`.

Die Ausgaben dieser beiden Befehle sind identisch und werden im Abschnitt [Ausgabeformat](#) beschrieben.

Cache-Engine-Version 1.4.14 oder höher

Für die Cache-Engine-Version 1.4.14 oder höher verwenden Sie den Befehl `config`. Dieser Befehl wurde den Memcached ASCII - und Binärprotokollen von hinzugefügt und ist im ElastiCache Cluster-Client implementiert. ElastiCache Wenn Sie Auto Discovery mit einer anderen Client-Bibliothek verwenden möchten, muss diese Bibliothek um den Support für den Befehl `config` erweitert werden.

Note

Die folgende Dokumentation bezieht sich auf ASCII das Protokoll. Der `config` Befehl unterstützt jedoch beide ASCII Protokolle und Binärdateien. Wenn Sie Auto Discovery-Unterstützung mithilfe des Binärprotokolls hinzufügen möchten, finden Sie weitere Informationen im [Quellcode für den ElastiCache Cluster-Client](#).

Syntax

```
config [sub-command] [key]
```

Optionen

Name	Beschreibung	Erforderlich
sub-command		Ja

Name	Beschreibung	Erforderlich
	Der Unterbefehl, der zum Interagieren mit einem Cache-Knoten verwendet wird. Für Auto Discovery lautet dieser Unterbefehl <code>get</code> .	
<code>key</code>	Der Schlüssel, mit dem die Cluster-Konfiguration gespeichert ist. Für Auto Discovery lautet dieser Schlüssel <code>cluster</code> .	Ja

Verwenden Sie zum Abrufen der Cluster-Konfigurationsinformationen den folgenden Befehl:

```
config get cluster
```

Cache-Engine-Version älter als 1.4.14

Verwenden Sie zum Abrufen der Cluster-Konfigurationsinformationen den folgenden Befehl:

```
get AmazonElastiCache:cluster
```

Note

Verändern Sie nicht den Schlüssel „:clusterAmazonElastiCache“, da sich dort die Cluster-Konfigurationsinformationen befinden. Wenn Sie diesen Schlüssel überschreiben, ist der Client möglicherweise für einen kurzen Zeitraum (nicht länger als 15 Sekunden) falsch konfiguriert, bevor die Konfigurationsinformationen ElastiCache automatisch und korrekt aktualisiert werden.

Ausgabeformat

Wenn Sie `config get cluster` oder `get AmazonElastiCache:cluster` verwenden, umfasst die Antwort zwei Zeilen:

- Die Versionsnummer der Konfigurationsinformationen. Jedes Mal, wenn ein Knoten hinzugefügt oder aus dem Cache-Cluster entfernt wird, wird die Versionsnummer um eins erhöht.
- Eine Liste der Cache-Knoten. Jeder Knoten in der Liste wird durch eine `hostname|ip-address|port-`Gruppe repräsentiert und durch ein Leerzeichen getrennt.

Am Ende jeder Zeile erscheint ein Wagenrücklauf- und Zeilenvorschubzeichen (CR + LF). Die Datenzeile enthält am Ende ein Zeilenvorschubzeichen (LF), dem die Zeichen CR + LF hinzugefügt werden. Die Config-Versionszeile wird durch LF ohne CR beendet.

Ein Cache-Cluster mit drei Knoten wird wie folgt dargestellt:

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Jeder Knoten wird sowohl mit der als auch mit der privaten IP-Adresse angezeigt. CNAME Die CNAME wird immer vorhanden sein. Wenn die private IP-Adresse nicht verfügbar ist, wird sie nicht angezeigt. Die Pipezeichen "|" werden jedoch trotzdem gedruckt.

Example

Das nachfolgende Beispiel zeigt die Nutzlast, die zurückgegeben wird, wenn Sie die Konfigurationsinformationen abfragen:

```
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

- Die zweite Zeile gibt an, dass die Konfigurationsinformationen bisher zwölfmal geändert wurden.
- In der dritten Zeile wird die Liste der Knoten in alphabetischer Reihenfolge nach Hostname angegeben. Diese Reihenfolge kann sich von der Sequenz unterscheiden, die Sie derzeit in Ihrer Client-Anwendung verwenden.

ElastiCache Kunden mit auto Erkennung

Cluster-Client-Programme können automatisch alle Cache-Cluster-Knoten identifizieren und eine Verbindung zu ihnen herstellen, auf denen die Memcached Engine ausgeführt wird.

In diesem Abschnitt wird die Installation und Konfiguration von und beschrieben. ElastiCache PHP NETClients zur Verwendung mit Auto Discovery.

Themen

- [Installieren und Kompilieren von Cluster-Clients](#)
- [Konfiguration von ElastiCache Clients](#)

Installieren und Kompilieren von Cluster-Clients

Dieser Abschnitt behandelt die Installation, Konfiguration und Kompilierung von PHP und NETCluster-Clients für ElastiCache auto Erkennung von Amazon.

Themen

- [Installation des ElastiCache Clusterclients für. NET](#)
- [Installieren des ElastiCache-Cluster-Clients-für-PHP](#)
- [Kompilieren des Quellcodes für den ElastiCache Cluster-Client für PHP](#)

Installation des ElastiCache Clusterclients für. NET

Du kannst das finden ElastiCache . NETCluster-Client-Code als Open Source unter <https://github.com/awslabs/elasticache-cluster-config-net>.

In diesem Abschnitt wird beschrieben, wie Sie den installieren, aktualisieren und entfernen. NETKomponenten für den ElastiCache Cluster-Client auf EC2 Amazon-Instances. Weitere Informationen zu Auto Discovery finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#). Zum Beispiel. NETCode zur Verwendung des Clients finden Sie unter [Verwenden des ElastiCache Cluster-Clients für. NET](#).

Themen

- [Installation. NET](#)
- [Laden Sie die herunter ElastiCache . NETCluster-Client für ElastiCache](#)
- [Installieren Sie AWS Baugruppen mit NuGet](#)

Installation. NET

Das musst du haben. NET3.5 oder höher installiert, um das verwenden zu können AWS . NETSDK für ElastiCache. Wenn du nicht hast. NET3.5 oder höher können Sie die neueste Version von <http://www.microsoft.com/net> herunterladen und installieren.

Laden Sie die herunter ElastiCache . NETCluster-Client für ElastiCache

Um das herunterzuladen ElastiCache . NETCluster-Client

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf ElastiCache Cluster Client.
3. Wählen Sie in der Liste ElastiCache Memcached Cluster Client herunterladen die Option aus. NET, und klicken Sie dann auf Herunterladen.

Installieren Sie AWS Baugruppen mit NuGet

NuGet ist ein Paketverwaltungssystem für die. NETPlattform. NuGet ist sich der Abhängigkeiten von Assemblys bewusst und installiert alle erforderlichen Dateien automatisch. NuGet installierte Assemblys werden zusammen mit Ihrer Lösung gespeichert und nicht etwa an einem zentralen OrtProgram Files, sodass Sie anwendungsspezifische Versionen installieren können, ohne dass Kompatibilitätsprobleme auftreten.

Installieren NuGet

NuGet kann über die Installationsgalerie unter installiert werdenMSDN; siehe <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>. Wenn Sie Visual Studio 2010 NuGet oder höher verwenden, wird es automatisch installiert.

Sie können entweder NuGet den Solution Explorer oder die Package Manager Console verwenden.

NuGet Vom Solution Explorer aus verwenden

Zur Verwendung NuGet im Solution Explorer in Visual Studio 2010

1. Wählen Sie im Menü Extras den Bibliotheks-Paket-Manager aus.
2. Klicken Sie auf Paket-Manager-Konsole.

Zur Verwendung NuGet aus dem Solution Explorer in Visual Studio 2012 oder Visual Studio 2013

1. Wählen Sie im Menü Tools die Option NuGet Package Manager aus.
2. Klicken Sie auf Paket-Manager-Konsole.

An der Befehlszeile können Sie die Komponenten mit Install-Package wie im Folgenden dargestellt installieren.

```
Install-Package Amazon.ElastiCacheCluster
```

Eine Seite für jedes Paket, das verfügbar ist, wie z. B. die Assemblys AWS SDK und AWS.Extensions, finden Sie auf der NuGet Website unter <http://www.nuget.org>. NuGet Die Seite für jedes Paket enthält eine Beispielbefehlszeile für die Installation des Pakets mithilfe der Konsole und eine Liste der früheren Versionen des Pakets, die über NuGet verfügbar sind.

Weitere Informationen zu den Befehlen der Paket-Manager-Konsole finden Sie unter <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>.

Installieren des ElastiCache-Cluster-Clients-für-PHP

In diesem Abschnitt wird beschrieben, wie Sie die PHP-Komponenten für den ElastiCache-Cluster-Client auf Amazon-EC2-Instances installieren, aktualisieren und entfernen. Weitere Informationen zu Auto Discovery finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#). Beispiele für PHP-Code zur Verwendung des Clients finden Sie unter [Verwenden des ElastiCache Cluster-Clients für PHP](#).

Themen

- [Herunterladen des Installationspakets](#)
- [Für Benutzer, die bereits die Erweiterung php-memcached installiert haben](#)
- [Installationsschritte für neue Benutzer](#)
- [Den PHP Cluster-Client entfernen](#)

Herunterladen des Installationspakets

Um sicherzustellen, dass Sie die richtige Version des ElastiCache Cluster-Clients für verwendenPHP, müssen Sie wissen, welche Version von auf Ihrer EC2 Amazon-Instance installiert PHP ist. Sie müssen auch wissen, ob auf Ihrer EC2 Amazon-Instance eine 64-Bit- oder 32-Bit-Version von Linux ausgeführt wird.

So ermitteln Sie die auf Ihrer EC2 Amazon-Instance installierte PHP Version

- Führen Sie in der Befehlszeile den folgenden Befehl aus:

```
php -v
```

Die PHP Version wird in der Ausgabe angezeigt, wie in diesem Beispiel:

```
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

Wenn Ihre Version PHP und die Memcached-Version nicht kompatibel sind, erhalten Sie eine Fehlermeldung, die etwa der folgenden ähnelt:

```
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 diesem Fall müssen Sie das Modul aus dem Quellcode kompilieren. Weitere Informationen finden Sie unter [Kompilieren des Quellcodes für den ElastiCache Cluster-Client für PHP](#).

Um Ihre EC2 AMI Amazon-Architektur zu ermitteln (64-Bit oder 32-Bit)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie in der Instance-Liste auf Ihre EC2 Amazon-Instance.
3. Suchen Sie auf der Registerkarte Beschreibung nach dem Feld AMI:. Die Beschreibung einer 64-Bit-Instance enthält x86_64, bei einer 32-Bit-Instance enthält dieses Feld i386 oder i686.

Sie sind jetzt bereit, den ElastiCache Cluster-Client herunterzuladen.

Um den ElastiCache Cluster-Client herunterzuladen für PHP

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der ElastiCache Konsole ElastiCache Cluster Client aus.
3. Wählen Sie aus der Liste ElastiCache Memcached Cluster Client herunterladen den ElastiCache Cluster-Client aus, der Ihrer PHP Version und AMI Architektur entspricht, und klicken Sie dann auf die Schaltfläche Herunterladen.

Für Benutzer, die bereits die Erweiterung php-memcached installiert haben

So aktualisieren Sie die **php-memcached**-Installation

1. Entfernen Sie die bisherige Installation der Memcached-Erweiterung für PHP, wie im Thema [Den PHP Cluster-Client entfernen](#) beschrieben.

2. Installieren Sie die neue ElastiCache-Erweiterung php-memcached wie zuvor in [Installationsschritte für neue Benutzer](#) beschrieben.

Installationsschritte für neue Benutzer

Themen

- [Installation von PHP 7.x für neue Benutzer](#)
- [Installation von PHP 5.x für neue Benutzer](#)

Installation von PHP 7.x für neue Benutzer

Themen

- [Um PHP 7 auf einem Ubuntu-Server 14.04 LTS AMI \(64-Bit und 32-Bit\) zu installieren](#)
- [So installieren Sie PHP 7 auf einem Amazon Linux 201609 AMI](#)
- [Um PHP 7 auf einem SUSE Linux zu installieren AMI](#)

Um PHP 7 auf einem Ubuntu-Server 14.04 LTS AMI (64-Bit und 32-Bit) zu installieren

1. Starten Sie eine neue Instanz von der. AMI
2. Führen Sie die folgenden Befehle aus:

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. Installieren Sie PHP 7.

```
sudo yum install php70
```

4. Laden Sie den Amazon ElastiCache Cluster Client herunter.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/
latest-64bit
```

5. Extrahieren Sie latest-64bit.

```
tar -zxvf latest-64bit
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` nach `/usr/lib/php/20151012`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. Fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei `/etc/php/7.0/cli/php.ini` ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php/7.0/cli/php.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 7 auf einem Amazon Linux 201609 AMI

1. Starten Sie eine neue Instance vom AMI
2. Führen Sie den folgenden Befehl aus:

```
sudo yum install gcc-c++
```

3. Installieren Sie PHP 7.

```
sudo yum install php70
```

4. Laden Sie den Amazon ElastiCache Cluster Client herunter.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. Extrahieren Sie `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` nach `/usr/lib64/php/7.0/modules/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. Erstellen Sie die Datei `50-memcached.ini`.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php-7.0.d/50-memcached.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Um PHP 7 auf einem SUSE Linux zu installieren AMI

1. Starten Sie eine neue Instanz vom AMI.
2. Führen Sie den folgenden Befehl aus:

```
sudo zypper install gcc
```

3. Installieren Sie PHP 7.

```
sudo yum install php70
```

4. Laden Sie den Amazon ElastiCache Cluster Client herunter.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. Extrahieren Sie `latest-64bit`.

```
tar -zxvf latest-64bit
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` nach `/usr/lib64/php7/extensions/`.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```

7. Fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei `/etc/php7/cli/php.ini` ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php7/cli/php.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Installation von PHP 5.x für neue Benutzer

Themen

- [Um PHP 5 auf einem Amazon Linux AMI 2014.03 \(64-Bit und 32-Bit\) zu installieren](#)
- [Um PHP 5 auf einem Red Hat Enterprise Linux 7.0 \(64-Bit und 32-Bit\) zu installieren AMI](#)
- [Um PHP 5 auf einem Ubuntu-Server 14.04 LTS AMI \(64-Bit und 32-Bit\) zu installieren](#)
- [Um PHP 5 für Linux Enterprise Server 11 \(64-Bit oder 32-Bit\) zu SUSE installieren AMI](#)
- [Andere Linux-Distributionen](#)

Um PHP 5 auf einem Amazon Linux AMI 2014.03 (64-Bit und 32-Bit) zu installieren

1. Starten Sie eine Amazon Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. Abhängigkeiten installierenPHP:

```
sudo yum install gcc-c++ php php-pear
```

3. Laden Sie das richtige `php-memcached` Paket für Ihre EC2 Amazon-Instance und PHP Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren `php-memcached`. Das URI sollte der Download-Pfad für das Installationspaket sein:

```
sudo pecl install <package download path>
```

Hier ist ein Beispiel für einen Installationsbefehl für PHP 5.4, 64-Bit-Linux. Ersetzen Sie in diesem Beispiel `X.Y.Z` durch die tatsächliche Versionsnummer:

```
sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```

 Note

Verwenden Sie immer die aktuelle Version des Installationsartefakts.

5. Fügen Sie mit root/sudo-Rechten eine neue Datei mit dem Namen `memcached.ini` im `/etc/php.d` Verzeichnis hinzu und fügen Sie „`extension=amazon-elasticache-cluster-client.so`“ in die Datei ein:

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Um PHP 5 auf einem Red Hat Enterprise Linux 7.0 (64-Bit und 32-Bit) zu installieren AMI

1. Starten Sie eine Red Hat Enterprise Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHPAbhängigkeiten installieren:

```
sudo yum install gcc-c++ php php-pear
```

3. Laden Sie das richtige `php-memcached` Paket für Ihre EC2 Amazon-Instance und PHP Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren `php-memcached`. Das URI sollte der Download-Pfad für das Installationspaket sein:

```
sudo pecl install <package download path>
```

5. Erstellen Sie mit Root-/Sudo-Berechtigung eine neue Datei namens `memcached.ini` im Verzeichnis `/etc/php.d` und fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/  
php.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Um PHP 5 auf einem Ubuntu-Server 14.04 LTS AMI (64-Bit und 32-Bit) zu installieren

1. Starten Sie eine Ubuntu Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. Abhängigkeiten installierenPHP:

```
sudo apt-get update  
sudo apt-get install gcc g++ php5 php-pear
```

3. Laden Sie das richtige php-memcached Paket für Ihre EC2 Amazon-Instance und PHP Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren php-memcached. Das URI sollte der Download-Pfad für das Installationspaket sein.

```
sudo pecl install <package download path>
```

Note

In diesem Installationsschritt wird das Build-Artefakt `amazon-elasticache-cluster-client.so` im Verzeichnis `/usr/lib/php5/20121212*` installiert. Überprüfen Sie den absoluten Pfad des Build-Artefakts, da er im nächsten Schritt benötigt wird.

Wenn der vorherige Befehl nicht funktioniert, müssen Sie das PHP Client-Artefakt manuell `amazon-elasticache-cluster-client.so` aus der heruntergeladenen `*.tgz` Datei extrahieren und in das `/usr/lib/php5/20121212*` Verzeichnis kopieren.

```
tar -xvf <package download path>  
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Fügen Sie mit root/sudo-Rechten eine neue Datei mit dem Namen `memcached.ini` im `/etc/php5/cli/conf.d` Verzeichnis hinzu und fügen Sie „`extension=<absolute path to .so>`“ in die Datei ein. `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. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Um PHP 5 für Linux Enterprise Server 11 (64-Bit oder 32-Bit) zu SUSE installieren AMI

1. Starten Sie eine SUSE Linux-Instanz (entweder 64-Bit oder 32-Bit) und melden Sie sich bei ihr an.
2. PHPAbhängigkeiten installieren:

```
sudo zypper install gcc php53-devel
```

3. Laden Sie das richtige `php-memcached` Paket für Ihre EC2 Amazon-Instanz und PHP Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren `php-memcached`. Das URI sollte der Download-Pfad für das Installationspaket sein.

```
sudo pecl install <package download path>
```

5. Erstellen Sie mit Root-/Sudo-Berechtigung eine neue Datei namens `memcached.ini` im Verzeichnis `/etc/php5/conf.d` und fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php5/conf.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Note

Sollte Schritt 5 für die vorherigen Plattformen nicht funktionieren, überprüfen Sie den Installationspfad für `amazon-elasticache-cluster-client.so`. Geben Sie auch den vollständigen Pfad der Binary in der Erweiterung an. Stellen Sie außerdem sicher, dass es sich PHP bei der verwendeten Version um eine unterstützte Version handelt. Die Versionen 5.3 bis 5.5 werden unterstützt.

Andere Linux-Distributionen

Auf einigen Systemen, insbesondere bei Cent OS7 und Red Hat Enterprise, wurde `libsas12.so.3` durch `libsas12.so.2` ersetzt. Wenn Sie auf diesen Systemen den ElastiCache Cluster-Client laden, versucht er, ihn zu finden und zu laden, aber er schlägt fehl. Um dieses Problem zu beheben, erstellen Sie einen symbolischen Link auf `libsas12.so.3`, damit der Client beim Laden von `libsas12.so.2` auf `libsas12.so.3` weitergeleitet wird. Mit dem folgenden Code wird dieser symbolische Link erstellt.

```
cd /usr/lib64
sudo ln libsas12.so.3 libsas12.so.2
```

Den PHP Cluster-Client entfernen

Themen

- [Eine frühere Version von PHP 7 wird entfernt](#)
- [Eine frühere Version von PHP 5 wird entfernt](#)

Eine frühere Version von PHP 7 wird entfernt

Um eine frühere Version von PHP 7 zu entfernen

1. Entfernen Sie die `amazon-elasticache-cluster-client.so` Datei aus dem entsprechenden Verzeichnis `PHP lib`, wie zuvor in den Installationsanweisungen angegeben. Weitere Informationen finden Sie in Installationsabschnitt unter [Für Benutzer, die bereits die Erweiterung `php-memcached` installiert haben](#).

2. Entfernen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` aus der Datei `php.ini`.
3. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Eine frühere Version von PHP 5 wird entfernt

Um eine frühere Version von PHP 5 zu entfernen

1. Entfernen Sie die Erweiterung `php-memcached`:

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Entfernen Sie die Datei `memcached.ini` aus dem entsprechenden Verzeichnis, wie zuvor in den Installationsanleitungen erläutert.

Kompilieren des Quellcodes für den ElastiCache Cluster-Client für PHP

In diesem Abschnitt wird beschrieben, wie Sie den Quellcode für den ElastiCache Cluster-Client für abrufen und kompilieren PHP.

[Es gibt zwei Pakete, aus denen Sie Daten abrufen GitHub und kompilieren müssen: aws-elasticache-cluster-client-libmemcached und -. aws-elasticache-cluster-client memcached-for-php](#)

Themen

- [Kompilieren der libmemcached-Bibliothek](#)
- [Kompilieren des ElastiCache Memcached Auto Discovery Clients für PHP](#)

Kompilieren der libmemcached-Bibliothek

Um die Bibliothek -libmemcached zu kompilieren aws-elasticache-cluster-client

1. Starten Sie eine EC2 Amazon-Instance.
2. Installieren Sie die Abhängigkeiten der Bibliothek.

- Auf Amazon Linux 201509 AMI

```
sudo yum install gcc gcc-c++ autoconf libevent-devel
```

- Auf Ubuntu 14.04 AMI

```
sudo apt-get update
sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev
```

3. Führen Sie einen Pull des Repositorys aus und kompilieren Sie den Code.

Download and install <https://github.com/aws-labs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz>

Kompilieren des ElastiCache Memcached Auto Discovery Clients für PHP

In den folgenden Abschnitten wird beschrieben, wie der ElastiCache Memcached Auto Discovery Client kompiliert wird

Themen

- [Kompilieren des ElastiCache Memcached-Clients für 7 PHP](#)
- [Den Memcached-Client für 5 kompilieren ElastiCache PHP](#)

Kompilieren des ElastiCache Memcached-Clients für 7 PHP

Führen Sie die folgenden Befehle im Code-Verzeichnis aus.

```
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

Sie können die Bibliothek libmemcached statisch mit der PHP Binärdatei verknüpfen, sodass sie auf verschiedene Linux-Plattformen portiert werden kann. Führen Sie dazu den folgenden Befehl vor make aus:

```
sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
```

Den Memcached-Client für 5 kompilieren ElastiCache PHP

Kompilieren Sie den `aws-elasticache-cluster-client-memcached-for-php`, indem Sie die folgenden Befehle im Ordner `aws-elasticache-cluster-client-memcached-for-php/` ausführen.

```
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
```

Konfiguration von ElastiCache Clients

Ein ElastiCache Cluster ist protokollkonform mit Valkey, Redis und Memcached. OSS Der Code, die Anwendungen und die beliebtesten Tools, die Sie heute in Ihrer bestehenden Umgebung verwenden, funktionieren problemlos mit dem Service.

In diesem Abschnitt werden spezifische Überlegungen zum Herstellen einer Verbindung zu Cache-Knoten in beschrieben ElastiCache.

Themen

- [Eingeschränkte Befehle](#)
- [Finden von Knotenendpunkten und Portnummern](#)
- [Verbinden für die Verwendung der automatischen Erkennung](#)
- [Verbindung zu Knoten in einem Valkey- oder OSS Redis-Cluster herstellen](#)
- [DNSNamen und zugrundeliegende IP](#)

Eingeschränkte Befehle

ElastiCache Schränkt den Zugriff auf bestimmte Cache-Engine-spezifische Befehle ein, die erweiterte Rechte erfordern, um ein Managed Service-Erlebnis zu bieten. Für Cache-Cluster, die auf Redis betrieben werden, sind folgende Befehle nicht verfügbar:

- `bgrewriteaof`
- `bgsave`
- `config`
- `debug`
- `migrate`
- `replicaof`
- `save`
- `slaveof`
- `shutdown`
- `sync`

Finden von Knotenendpunkten und Portnummern

Um eine Verbindung zu einem Cache-Knoten herzustellen, muss Ihre Anwendung den Endpunkt und die Portnummer für diesen Knoten kennen.

Finden von Knotenendpunkten und Portnummern (Konsole)

So bestimmen Sie Knotenendpunkte und Portnummern

1. Melden Sie sich bei der [Amazon ElastiCache Management Console](#) an und wählen Sie die Engine aus, die auf Ihrem Cluster läuft.

Es wird eine Liste aller Cluster der ausgewählten Engine angezeigt.

2. Fahren Sie mit der Anleitung für die ausgewählte Engine und Konfiguration fort.
3. Wählen Sie den Namen des gewünschten Clusters aus.
4. Entnehmen Sie die benötigten Informationen für den gewünschten Knoten aus den Spalten Port und Endpoint.

Finden von Cache-Knotenendpunkten und Portnummern (AWS CLI)

Verwenden Sie den Befehl `describe-cache-clusters` mit dem Parameter `--show-cache-node-info`, um die Endpunkte und Portnummern von Cache-Knoten zu bestimmen.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

Die vollqualifizierten DNS Namen und Portnummern befinden sich im Abschnitt Endpoint der Ausgabe.

Suchen nach Endpunkten und Portnummern von Cache-Knoten () ElastiCache API

Verwenden Sie die Aktion `DescribeCacheClusters` mit dem Parameter `ShowCacheNodeInfo=true`, um die Endpunkte und Portnummern von Cache-Knoten zu bestimmen.

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
```

Verbinden für die Verwendung der automatischen Erkennung

Wenn Sie in Ihren Anwendungen Auto Discovery verwenden, müssen Sie anstelle der einzelnen Endpunkte für jeden Cache-Knoten den Konfigurationsendpunkt für das Cluster kennen. Weitere Informationen finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#).

Note

Auto Discovery ist derzeit nur für Cache-Cluster verfügbar, die in Memcached ausgeführt werden.

Verbindung zu Knoten in einem Valkey- oder OSS Redis-Cluster herstellen

Note

Derzeit werden Cluster (API/CLI: Replikationsgruppen), die Replikation und Read Replicas unterstützen, nur für Cluster unterstützt, auf denen Valkey oder Redis ausgeführt werden.
OSS

ElastiCache Stellt für Cluster Konsole und API Schnittstellen zum Abrufen von Verbindungsinformationen für einzelne Knoten bereit. CLI

Für reine Lesezugriffe können Anwendungen eine Verbindung zu beliebigen Knoten im Cluster herstellen. Für Schreibaktivitäten empfehlen wir jedoch, dass Ihre Anwendungen eine Verbindung zum primären Endpunkt (Valkey oder Redis OSS (Clustermodus deaktiviert)) oder Konfigurationsendpunkt (Valkey oder Redis OSS (Clustermodus aktiviert)) für den Cluster herstellen, anstatt eine direkte Verbindung zu einem Knoten herzustellen. Dadurch wird sichergestellt, dass Ihre

Anwendungen immer den korrekten Knoten finden, auch wenn Sie das Cluster neu konfigurieren und einem Lesereplikat die primäre Rolle zuweisen.

Verbinden mit Clustern in einer Replikationsgruppe (Konsole)

So bestimmen Sie Endpunkte und Portnummern

- Weitere Informationen finden Sie auch im Thema [Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Verbinden mit Clustern in einer Replikationsgruppe (AWS CLI)

So bestimmen Sie die Cache-Knoten-Endpunkte und Portnummern

Verwenden Sie den Befehl `describe-replication-groups` mit dem Namen Ihrer Replikationsgruppe:

```
aws elasticache describe-replication-groups redis2x2
```

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

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

```

Verbindung zu Clustern in einer Replikationsgruppe herstellen () ElastiCache API

So bestimmen Sie die Cache-Knoten-Endpunkte und Portnummern

Rufen Sie DescribeReplicationGroups mit folgendem Parameter auf:

ReplicationGroupId = der Name Ihrer Replikationsgruppe

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>
```

DNSNamen und zugrundeliegende IP

Clients verfügen über eine Serverliste mit den Adressen und Ports der Server, auf denen Cache-Daten abgelegt werden. Bei der Verwendung ElastiCache gibt das DescribeCacheClusters API (oder das describe-cache-clusters Befehlszeilenprogramm) einen vollqualifizierten DNS Eintrag und eine Portnummer zurück, die für die Serverliste verwendet werden können.

Important

Es ist wichtig, dass Client-Anwendungen so konfiguriert sind, dass sie häufig DNS Namen von Cache-Knoten auflösen, wenn sie versuchen, eine Verbindung zu einem Cache-Knoten-Endpoint herzustellen.

VPCInstallationen

ElastiCache stellt sicher, dass sowohl der DNS Name als auch die IP-Adresse des Cache-Knotens unverändert bleiben, wenn Cache-Knoten im Falle eines Fehlers wiederhergestellt werden.

Nicht- VPC Installationen

ElastiCache stellt sicher, dass der DNS Name eines Cache-Knotens unverändert bleibt, wenn Cache-Knoten im Falle eines Fehlers wiederhergestellt werden. Die zugrunde liegende IP-Adresse des Cache-Knotens kann sich jedoch ändern.

Persistente Cache-Knotenverbindungen werden von den meisten Client-Bibliotheken standardmäßig unterstützt. Wir empfehlen, bei der Verwendung persistente Cache-Knotenverbindungen zu verwenden ElastiCache. Das clientseitige DNS Caching kann an mehreren Stellen erfolgen, z. B. in Clientbibliotheken, in der Sprachlaufzeit oder im Client-Betriebssystem. Prüfen Sie die Konfiguration Ihrer Anwendung auf allen Ebenen, um sicherzustellen, dass IP-Adressen für Ihre Cache-Knoten regelmäßig aufgelöst werden.

Daten-Tiering ElastiCache

ElastiCache Bei Valkey- oder OSS Redis-Clustern, die eine Replikationsgruppe bilden und einen Knotentyp aus der R6GD-Familie verwenden, werden die Daten zwischen Arbeitsspeicher und lokalem Speicher SSD (Solid-State-Laufwerke) aufgeteilt. Data Tiering bietet eine neue Preis-Leistungs-Option für Valkey- oder OSS Redis-Workloads, da neben der Speicherung von Daten im Arbeitsspeicher auch kostengünstigere Solid-State-Laufwerke (SSDs) in jedem Clusterknoten verwendet werden. Es ist ideal für Workloads, die regelmäßig auf bis zu 20 Prozent ihres gesamten Datensatzes zugreifen, sowie für Anwendungen, die zusätzliche Latenz beim Datenzugriff tolerieren können. SSD

ElastiCache Überwacht auf ElastiCache Clustern mit Daten-Tiering die letzte Zugriffszeit jedes gespeicherten Elements. Wenn der verfügbare Speicher (DRAM) vollständig verbraucht ist, verwendet er einen Algorithmus, der zuletzt ElastiCache verwendet wurde (LRU), um Objekte, auf die selten zugegriffen wird, automatisch aus dem Speicher in den Speicher zu verschieben. SSD Wenn anschließend auf Daten zugegriffen SSD wird, werden sie ElastiCache automatisch und asynchron zurück in den Arbeitsspeicher verschoben, bevor die Anforderung verarbeitet wird. Wenn Sie eine Workload haben, die regelmäßig nur auf eine Teilmenge ihrer Daten zugreift, ist Daten-Tiering eine optimale Möglichkeit, Ihre Kapazität kostengünstig zu skalieren.

Beachten Sie, dass bei der Verwendung von Data Tiering die Schlüssel selbst immer im Speicher verbleiben, während der Wert die Platzierung der Werte LRU im Speicher und nicht auf der Festplatte bestimmt. Im Allgemeinen empfehlen wir, dass Ihre Schlüsselgrößen kleiner als Ihre Wertgrößen sind, wenn Sie Daten-Tiering verwenden.

Das Daten-Tiering ist so konzipiert, dass es minimale Auswirkungen auf die Leistung von Anwendungs-Workloads hat. Wenn Sie beispielsweise von 500-Byte-Zeichenkettenwerten ausgehen,

können Sie bei Anfragen an Daten, die auf gespeichert sind, im Durchschnitt mit einer zusätzlichen Latenz von 300 Mikrosekunden SSD im Vergleich zu Anforderungen an Daten im Speicher rechnen.

Mit der größten Daten-Tiering-Knotengröße (cache.r6gd.16xlarge) können Sie bis zu 1 Petabyte Daten in einem einzigen Cluster mit 500 Knoten speichern (500 TB bei Verwendung von 1 Lesereplikat). Data Tiering ist mit allen Valkey- oder OSS Redis-Befehlen und Datenstrukturen kompatibel, die in unterstützt werden. ElastiCache Um diese Funktion nutzen zu können, sind keine clientseitigen Änderungen erforderlich.

Themen

- [Bewährte Methoden](#)
- [Einschränkungen](#)
- [Preisgestaltung](#)
- [Überwachen](#)
- [Verwenden von Daten-Tiering](#)
- [Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering](#)

Bewährte Methoden

Wir empfehlen Ihnen, die folgenden bewährten Methoden:

- Data Tiering ist ideal für Workloads, die regelmäßig auf bis zu 20 Prozent ihres gesamten Datensatzes zugreifen, sowie für Anwendungen, die zusätzliche Latenz beim Datenzugriff tolerieren können. SSD
- Wenn Sie die SSD Kapazität verwenden, die auf Knoten mit Datenschicht verfügbar ist, empfehlen wir, dass die Wertgröße größer als die Schlüsselgröße ist. Wenn Elemente zwischen DRAM und verschoben werden SSD, bleiben die Schlüssel immer im Speicher und nur die Werte werden in die SSD Ebene verschoben.

Einschränkungen

Für Daten-Tiering gelten die folgenden Beschränkungen:

- Sie können Daten-Tiering nur für Cluster verwenden, die Teil einer Replikationsgruppe sind.
- Der verwendete Knotentyp muss aus der r6gd-Familie stammen, die in den folgenden Regionen verfügbar ist: us-east-2, us-east-1, us-west-2, us-west-1, eu-west-1, eu-central-1,

Wenn die konfigurierte Löschrictlinie dies zulässt, ElastiCache wird mit dem Löschen von Elementen begonnen, wenn der Prozentsatz der Elemente im Speicher unter 5 Prozent sinkt. Auf Knoten, für die die Noeviction-Richtlinie konfiguriert wurde, wird bei Schreibvorgängen der Fehler „Nicht genügend Arbeitsspeicher“ angezeigt.

Es wird dennoch empfohlen, die Skalierung für Cluster mit aktiviertem Clustermodus oder die Hochskalierung für Cluster mit deaktiviertem Clustermodus in Betracht zu ziehen, wenn der Prozentsatz der Elemente im Arbeitsspeicher unter 5 Prozent sinkt. Weitere Informationen zur Skalierung finden Sie unter [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#). Weitere Informationen zu Metriken für Valkey- oder OSS Redis-Cluster, die Data Tiering verwenden, finden Sie unter [Metriken für Valkey und Redis OSS](#)

Verwenden von Daten-Tiering

Verwenden von Data Tiering mit dem AWS Management Console

Wenn Sie einen Cluster als Teil einer Replikationsgruppe erstellen, verwenden Sie Daten-Tiering, indem Sie einen Knotentyp aus der R6gd-Familie auswählen, z. B. cache.r6gd.xlarge. Bei Auswahl dieses Knotentyps wird das Daten-Tiering automatisch aktiviert.

Weitere Informationen zum Erstellen von Clustern finden Sie unter [Einen Cluster für Valkey oder Redis erstellen OSS](#).

Aktivieren von Data Tiering mit dem AWS CLI

Beim Erstellen einer Replikationsgruppe mithilfe von verwenden Sie Data Tiering AWS CLI, indem Sie einen Knotentyp aus der r6gd-Familie auswählen, z. B. cache.r6gd.xlarge, und den Parameter festlegen. `--data-tiering-enabled`

Sie können sich das Daten-Tiering nicht abwählen, wenn Sie einen Knotentyp aus der R6gd-Familie auswählen. Wenn Sie den Parameter `--no-data-tiering-enabled` festlegen, schlägt die Operation fehl.

Für Linux, macOS oder 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 \  
  --data-tiering-enabled
```

```
--cache-node-type cache.r6gd.xlarge \  
--engine redis \  
--cache-subnet-group-name default \  
--automatic-failover-enabled \  
--data-tiering-enabled
```

Für 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
```

Nach dem Ausführen dieses Vorgangs sehen Sie eine Antwort ähnlich dem folgenden:

```
{  
  "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  
  }  
}
```

Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering

Sie können ein Backup auf einem neuen Cluster mit aktiviertem Data Tiering mithilfe von (Console), () oder () wiederherstellen. AWS CLI ElastiCache API Wenn Sie einen Cluster mit Knotentypen in der R6gd-Familie erstellen, ist Daten-Tiering aktiviert.

Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering (Konsole)

So stellen Sie ein Backup in einem neuen Cluster mit aktiviertem Daten-Tiering wieder her (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, aus der Sie wiederherstellen möchten.
4. Wählen Sie Restore (Wiederherstellen) aus.
5. Füllen Sie das Dialogfeld Restore Cluster aus. Füllen Sie unbedingt alle Pflichtfelder aus sowie alle anderen, deren Standardeinstellungen Sie ändern möchten.
 1. Cluster-ID – Erforderlich. Der Name des neuen Clusters.
 2. Clustermodus aktiviert (Scale Out) — Wählen Sie diese Option für einen Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert).
 3. Node Type (Knotentyp) – Geben Sie cache.r6gd.xlarge oder einen anderen Knotentyp aus der R6gd-Familie an.
 4. Anzahl der Shards — Wählen Sie die Anzahl der Shards aus, die Sie im neuen Cluster haben möchten (API/CLI: Knotengruppen).
 5. Replicas pro Shard – Wählen Sie die Anzahl der Lesereplikat-Knoten aus, die sich in jedem Shard befinden sollen.
 6. Slots und Keyspaces – Wählen Sie aus, wie die Schlüssel auf die Shards verteilt werden sollen. Wenn Sie die Schlüsselverteilungen angeben möchten, füllen Sie die Tabelle mit den Schlüsselbereichen für jeden Shard aus.
 7. Availability zone(s) – Geben Sie an, wie die Availability Zones des Clusters ausgewählt werden sollen.
 8. Port – Ändern Sie diese Einstellung nur, wenn der neue Cluster einen anderen Port verwenden soll.

9. Wählen Sie einen VPC — Wählen Sie den aus, VPC in dem dieser Cluster erstellt werden soll.
 - 10 Parametergruppe — Wählen Sie eine Parametergruppe, die ausreichend Speicher für den Valkey- oder OSS Redis-Overhead für den ausgewählten Knotentyp reserviert.
6. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Erstellen.

Weitere Informationen zum Erstellen von Clustern finden Sie unter [Einen Cluster für Valkey oder Redis erstellen OSS](#).

Wiederherstellen von Daten aus einem Backup in Cluster mit aktiviertem Daten-Tiering (AWS CLI)

Beim Erstellen einer Replikationsgruppe mithilfe von wird Data Tiering standardmäßig verwendet AWS CLI, indem ein Knotentyp aus der r6gd-Familie ausgewählt wird, z. B. cache.r6gd.xlarge, und der Parameter festgelegt wird. `--data-tiering-enabled`

Sie können sich das Daten-Tiering nicht abwählen, wenn Sie einen Knotentyp aus der R6gd-Familie auswählen. Wenn Sie den Parameter `--no-data-tiering-enabled` festlegen, schlägt die Operation fehl.

Für Linux, macOS oder 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
```

Für Linux, macOS oder 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
```

Nach dem Ausführen dieses Vorgangs sehen Sie eine Antwort ähnlich dem folgenden:

```
{
  "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
  }
}
```

Einen Cluster vorbereiten in ElastiCache

Im Folgenden finden Sie Anweisungen zum Erstellen eines Clusters mithilfe der ElastiCache Konsole, AWS CLI, oder der ElastiCache API.

Sie können einen ElastiCache Cluster auch mithilfe von erstellen [AWS CloudFormation](#). Weitere Informationen finden Sie unter [AWS:ElastiCache::: CacheCluster](#) im AWS Cloud Formation-Benutzerhandbuch, das Anleitungen zur Implementierung dieses Ansatzes enthält.

Wenn Sie einen Cluster oder eine Replikationsgruppe erstellen, ist es ratsam, einige Vorbereitungen zu treffen, damit Sie nicht sofort ein Upgrade durchführen oder Änderungen vornehmen müssen.

Themen

- [Ermitteln Ihrer ElastiCache Cluster-Anforderungen](#)
- [Auswahl der Knotengröße](#)

Ermitteln Ihrer ElastiCache Cluster-Anforderungen

Vorbereitung

Wenn Sie die Antworten auf die folgenden Fragen kennen, kann die ElastiCache Clustererstellung reibungsloser vonstatten gehen:

- Welchen Knoten-Instance-Typ benötigen Sie?

Eine Anleitung bei der Auswahl eines Knoten-Instance-Typs finden Sie unter [Auswahl der Knotengröße](#).

- Werden Sie Ihren Cluster in einer virtuellen privaten Cloud (VPC) auf Basis von Amazon startenVPC?

Important

Wenn Sie Ihren Cluster in einem starten möchten, stellen Sie sicherVPC, dass Sie dort eine Subnetzgruppe erstellen, VPC bevor Sie mit der Erstellung eines Clusters beginnen. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).

ElastiCache ist so konzipiert, dass der Zugriff AWS über Amazon von innen möglich istEC2. Wenn Sie jedoch in einem auf Amazon VPC basierenden System starten VPC und sich Ihr Cluster in einem befindetVPC, können Sie den Zugriff von außen gewähren AWS. Weitere Informationen finden Sie unter [Zugriff auf ElastiCache Ressourcen von außen AWS](#).

- Müssen Sie irgendwelche Parameterwerte anpassen?

Erstellen Sie in diesem Fall eine benutzerdefinierte Parametergruppe. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

Wenn Sie Valkey oder Redis verwendenOSS, sollten Sie die Einstellung `reserved-memory` von oder in Betracht ziehen. `reserved-memory-percent` Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

- Müssen Sie Ihre eigene VPCSicherheitsgruppe erstellen?

Weitere Informationen finden Sie unter [Sicherheit in Ihrem VPC](#).

- Haben Sie vor, Fehlertoleranz zu implementieren?

Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Themen

- [ElastiCache Speicher- und Prozessoranforderungen](#)
- [Konfiguration von Memcached-Cluster](#)
- [Valkey- und Redis-Clusterkonfiguration OSS](#)
- [ElastiCache Anforderungen an die Skalierung](#)
- [ElastiCache Zugriffsanforderungen](#)
- [Anforderungen an Region, Availability Zone und Lokale Zone für ElastiCache](#)

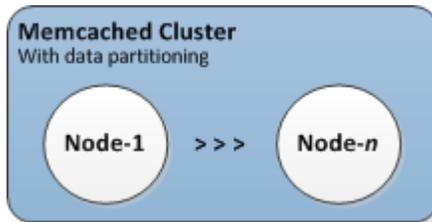
ElastiCache Speicher- und Prozessoranforderungen

Der Grundbaustein von Amazon ElastiCache ist der Node. Knoten werden zur Bildung von Clustern einzeln oder in Gruppierungen konfiguriert. Berücksichtigen Sie bei der Bestimmung des für den Cluster zu verwendenden Knotentyps die Knotenkonfiguration des Clusters und die zu speichernde Datenmenge.

Die Memcached-Engine arbeitet mit mehreren Threads. Daher wirkt sich die Anzahl der Kerne eines Knotens auf die für den Cluster verfügbare Rechenleistung aus.

Konfiguration von Memcached-Cluster

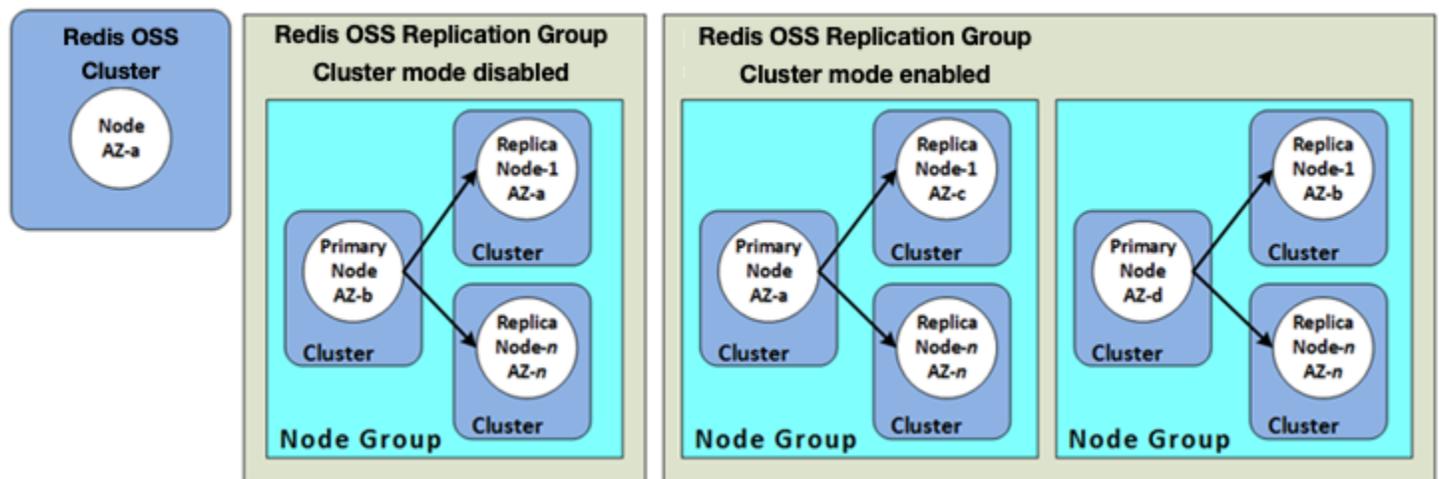
ElastiCache (Memcached-) Cluster bestehen aus 1 bis 60 Knoten. Die Daten in einem Memcached-Cluster werden über Knoten im Cluster hinweg partitioniert. Ihre Anwendung stellt über eine als Endpunkt bezeichnete Netzwerkadresse eine Verbindung mit einem Memcached-Cluster her. Jeder Knoten in einem Memcached-Cluster verfügt über seinen eigenen Endpunkt. Dieser wird von der Anwendung zum Lesen aus oder Schreiben in einem bestimmten Knoten genutzt. Zusätzlich zu den Knotenendpunkten hat der Memcached-Cluster selbst einen Endpunkt, der als Konfigurationsendpunkt bezeichnet wird. Ihre Anwendung kann diesen Endpunkt verwenden, um aus dem Cluster zu lesen oder in den Cluster zu schreiben, wobei die Bestimmung des Knotens, aus dem gelesen oder in den geschrieben werden soll, Auto Discovery überlassen bleibt.



Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

Valkey- und Redis-Clusterkonfiguration OSS

ElastiCache Bei Valkey und Redis bestehen OSS Cluster aus 0 bis 500 Shards (auch Knotengruppen genannt). Die Daten in einem Valkey- oder OSS Redis-Cluster sind auf die Shards im Cluster verteilt. Ihre Anwendung stellt über eine Netzwerkadresse, die als Endpunkt bezeichnet wird, eine Verbindung zu einem Valkey- oder OSS Redis-Cluster her. Die Knoten in einem Valkey- oder OSS Redis-Shard erfüllen eine von zwei Rollen: ein primärer Knoten mit Lese-/Schreibzugriff und alle anderen Knoten sind nur lesbare Sekundärknoten (auch Read Replicas genannt). Zusätzlich zu den Knotenendpunkten verfügt der Valkey- oder Redis-Cluster selbst über einen Endpunkt, den sogenannten Konfigurationsendpunkt. OSS Ihre Anwendung kann diesen Endpunkt verwenden, um aus dem Cluster zu lesen oder in ihn zu schreiben, sodass die Entscheidung, von welchem Knoten gelesen oder auf welchen geschrieben werden soll, überlassen bleibt ElastiCache (Redis). OSS



Weitere Informationen finden Sie unter [Verwaltung von Clustern in ElastiCache](#).

ElastiCache Anforderungen an die Skalierung

Alle Cluster können hochskaliert werden, indem ein neuer Cluster mit einem neuen, größeren Knotentyp erstellt wird. Wenn Sie einen Memcached-Cluster hochskalieren, ist der neue Cluster zunächst leer. Wenn Sie einen Valkey- oder OSS Redis-Cluster hochskalieren, können Sie ihn von einem Backup aus starten und so verhindern, dass der neue Cluster leer anfängt.

Amazon ElastiCache for Memcached-Cluster können horizontal oder horizontal skaliert werden. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten aus dem Cluster nach oben bzw. nach unten skaliert. Wenn Automatic Discovery aktiviert ist und die Anwendung mit dem Konfigurationsendpunkt des Clusters verbunden ist, sind zum Hinzufügen oder Entfernen von Knoten keine Änderungen an Ihrer Anwendung erforderlich.

Weitere Informationen finden Sie unter [Skalierung ElastiCache](#) in diesem Handbuch.

ElastiCache Zugriffsanforderungen

Standardmäßig wird auf ElastiCache Amazon-Cluster von EC2 Amazon-Instances aus zugegriffen. Der Netzwerkzugriff auf einen ElastiCache Cluster ist auf das Konto beschränkt, mit dem der Cluster erstellt wurde. Bevor Sie von einer EC2 Amazon-Instance aus auf einen Cluster zugreifen können, müssen Sie daher die EC2 Amazon-Instance autorisieren, auf den Cluster zuzugreifen. Die dazu erforderlichen Schritte variieren, je nachdem, ob Sie Classic VPC oder Classic EC2 gestartet haben EC2.

Wenn Sie Ihren Cluster in gestartet EC2 haben, müssen VPC Sie dem Cluster Netzwerkzugang gewähren. Wenn Sie Ihren Cluster in EC2 -Classic gestartet haben, müssen Sie der Amazon Elastic Compute Cloud-Sicherheitsgruppe, die mit der Instance verknüpft ist, Zugriff auf Ihre ElastiCache Sicherheitsgruppe gewähren. Detaillierte Anweisungen finden Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) in diesem Handbuch.

Anforderungen an Region, Availability Zone und Lokale Zone für ElastiCache

Amazon ElastiCache unterstützt alle AWS Regionen. Indem Sie Ihre ElastiCache Cluster in einer AWS Region in der Nähe Ihrer Anwendung platzieren, können Sie die Latenz reduzieren. Wenn Ihr Cluster aus mehreren Knoten besteht, können Sie die Auswirkungen von Ausfällen auf Ihren Cluster verringern, indem Sie Ihre Knoten in verschiedenen Availability Zones oder in Local Zones unterbringen.

Weitere Informationen finden Sie hier:

- [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#)
- [Verwenden von lokalen Zonen mit ElastiCache](#)
- [Minimieren von Ausfällen](#)

Auswahl der Knotengröße

Die Knotengröße, die Sie für Ihren ElastiCache Cluster auswählen, wirkt sich auf Kosten, Leistung und Fehlertoleranz aus.

Knotengröße (Valkey und OSS Redis)

Informationen zu den Vorteilen von Graviton-Prozessoren finden Sie unter [AWS -Graviton-Prozessor](#).

Die Beantwortung der folgenden Fragen kann Ihnen helfen, den minimalen Knotentyp zu ermitteln, den Sie für Ihre Valkey- oder Redis-Implementierung benötigen: OSS

- Erwarten Sie durchsatzgebundene Workloads mit mehreren Client-Verbindungen?

Wenn dies der Fall ist und Sie Redis OSS Version 5.0.6 oder höher verwenden, können Sie mit unserer erweiterten I/O-Funktion, die, sofern verfügbar, für die Entlastung der Client-Verbindungen im Namen der Redis-Engine CPUs verwendet wird, einen besseren Durchsatz und eine bessere Latenz erzielen. OSS Wenn Sie Redis OSS Version 7.0.4 oder höher verwenden, erhalten Sie zusätzlich zur erweiterten I/O zusätzliche Beschleunigung durch erweitertes I/O-Multiplexing, bei dem jeder dedizierte Netzwerk-I/O-Thread Befehle von mehreren Clients an die OSS Redis-Engine weiterleitet und so die Fähigkeit von Redis OSS nutzt, Befehle effizient stapelweise zu verarbeiten. In ElastiCache (RedisOSS) v7.1 und höher haben wir die erweiterte I/O-Thread-Funktionalität erweitert, um auch die Logik der Präsentationsebene zu handhaben. Mit Darstellungsschicht meinen wir, dass erweiterte I/O-Threads jetzt nicht nur die Client-Eingaben lesen, sondern die Eingabe auch im OSS Redis-Binärbefehlsformat parsen, die dann zur Ausführung an den Haupt-Thread weitergeleitet werden, was zu einer Leistungssteigerung führt. Weitere Informationen finden Sie im [Blogbeitrag](#) und auf der Seite mit den [unterstützten Versionen](#).

- Haben Sie Workloads, die regelmäßig auf einen kleinen Prozentsatz ihrer Daten zugreifen?

Wenn dies der Fall ist und Sie die OSS Redis-Engine Version 6.2 oder höher verwenden, können Sie das Data-Tiering nutzen, indem Sie den Knotentyp r6gd wählen. Beim Data-Tiering werden Daten, die zuletzt verwendet wurden, in gespeichert. SSD Beim Abruf entsteht eine geringe Latenzzeit, die jedoch durch Kosteneinsparungen ausgeglichen wird. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Wie viel Gesamtspeicher benötigen Sie für Ihre Daten?

Um eine allgemeine Schätzung zu erhalten, nehmen Sie die Größe der Objekte, die Sie zwischenspeichern möchten. Multiplizieren Sie diese Größe mit der Anzahl der Objekte, die Sie gleichzeitig im Cache halten wollen. Um eine vernünftige Schätzung der Elementgröße zu erhalten, serialisieren Sie zunächst Ihre Cache-Elemente und zählen dann die Zeichen. Teilen Sie diese Zahl dann durch die Anzahl der Shards in Ihrem Cluster.

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Welche Version von OSS Redis verwenden Sie?

Bei OSS Redis-Versionen vor 2.8.22 müssen Sie mehr Speicher für Failover, Snapshots, Synchronisation und das Heraufstufen eines Replikats zu primären Vorgängen reservieren. Diese Anforderung besteht, weil ausreichend Speicher für alle Schreibvorgänge während des Prozesses zur Verfügung stehen muss.

OSSRedis-Version 2.8.22 und höher verwenden einen Forkless-Speicherprozess, der weniger verfügbaren Speicher benötigt als der vorherige Prozess.

Weitere Informationen finden Sie hier:

- [So werden Synchronisation und Backup implementiert](#)
 - [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#)
- Wie hoch ist der Anteil der Schreibvorgänge Ihrer Anwendung?

Anwendungen mit vielen Schreibvorgängen können deutlich mehr verfügbaren Speicher, d. h. Speicher, der nicht von Daten belegt wird, beim Erstellen von Snapshots oder beim Failover erfordern. Wenn der BGSAVE-Prozess ausgeführt wird, müssen Sie über genügend Speicher verfügen, der nicht durch Daten belegt ist, um alle Schreibvorgänge, die während des BGSAVE-Prozesses stattfinden, unterzubringen. Beispiele hierfür sind das Erstellen eines Snapshots, das Synchronisieren eines primären Clusters mit einem Replikat in einem Cluster und das Aktivieren der Funktion „Nur Datei anhängen“ (). AOF Ein weiteres ist das Heraufstufen eines Replikats zum primären Knoten (wenn Sie Multi-AZ aktiviert haben). Im schlimmsten Fall werden alle Ihre Daten während des Prozesses neu geschrieben. In diesem Fall benötigen Sie eine Knoten-Instance-Größe mit doppelt so viel Speicher wie für die Daten allein benötigt wird.

Detailliertere Informationen erhalten Sie unter [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#).

- Handelt es sich bei Ihrer Implementierung um einen eigenständigen Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) oder um einen Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert) mit mehreren Shards?

Valkey- oder OSS Redis-Cluster (Cluster-Modus deaktiviert)

Wenn Sie einen Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) implementieren, muss Ihr Knotentyp in der Lage sein, all Ihre Daten sowie den erforderlichen Overhead aufzunehmen, wie im vorherigen Punkt bullet.

Nehmen Sie zum Beispiel an, dass die Gesamtgröße aller Ihrer Objekte 12 GB beträgt. In diesem Fall können Sie einen `cache.m3.xlarge`-Knoten mit 13,3 GB Speicher oder einen `cache.r3.large`-Knoten mit 13,5 GB Speicher verwenden. Sie benötigen jedoch möglicherweise mehr Speicher für BGSAVE-Operationen. Wenn Ihre Anwendung sehr schreibintensiv ist, sollten Sie den Speicherbedarf auf mindestens 24 GB verdoppeln. Verwenden Sie also entweder eine `cache.m3.2xlarge` mit 27,9 GB Speicher oder eine `cache.r3.xlarge` mit 30,5 GB Speicher.

Valkey oder Redis OSS (Cluster-Modus aktiviert) mit mehreren Shards

Wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mit mehreren Shards implementieren, muss der Knotentyp Datenbytes aufnehmen können. `bytes-for-data-and-overhead / number-of-shards`

Angenommen, Sie schätzen die Gesamtgröße aller Ihrer Objekte auf 12 GB und Sie haben zwei Shards. In diesem Fall können Sie einen `cache.m3.large`-Knoten mit 6,05 GB Speicher (12 GB / 2) verwenden. Sie benötigen jedoch möglicherweise mehr Speicher für BGSAVE-Operationen. Wenn Ihre Anwendung schreibintensiv ist, verdoppeln Sie die Speicheranforderungen auf mindestens 12 GB pro Shard. Verwenden Sie also entweder eine `cache.m3.xlarge` mit 13,3 GB Speicher oder eine `cache.r3.large` mit 13,5 GB Speicher.

- Verwenden Sie Local Zones?

[Local Zones](#) ermöglichen es Ihnen, Ressourcen wie einen ElastiCache Cluster an mehreren Standorten in der Nähe Ihrer Benutzer zu platzieren. Bei der Auswahl der Knotengröße sollten Sie jedoch beachten, dass die verfügbaren Knotengrößen unabhängig von den Kapazitätsanforderungen derzeit auf die folgenden beschränkt sind:

- Aktuelle Generation:

M5-Knotentypen: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`,
`cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

R5-Knotentypen: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`,
`cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

T3-Knotentypen: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Während Ihr Cluster läuft, können Sie die Messwerte für Speicherverbrauch, Prozessorauslastung, Cache-Treffer und Cache-Fehlschläge überwachen, die veröffentlicht werden CloudWatch. Sie werden vielleicht feststellen, dass Ihr Cluster nicht die gewünschte Trefferquote hat oder dass die Schlüssel zu oft entfernt werden. In diesen Fällen können Sie eine andere Knotengröße mit größeren CPU und Speicherspezifikationen wählen.

Denken Sie bei der Überwachung der CPU Nutzung daran, dass Valkey und Redis OSS Single-Threading verwenden. Multiplizieren Sie also die gemeldete CPU Nutzung mit der Anzahl der CPU Kerne, um die tatsächliche Nutzung zu ermitteln. Beispielsweise ist ein Vierkernprozessor, der eine Nutzungsrate von 20 Prozent CPU meldet, tatsächlich der Einkernprozessor, der bei einer Auslastung von 80 Prozent OSS läuft.

Knotengröße (Memcached)

Memcached-Cluster enthalten gegebenenfalls mehrere Knoten mit über die Knoten partitionierten Cluster-Daten. Aus diesem Grund sind die Speicheranforderungen des Clusters und die eines Knotens zwar ähnlich, jedoch nicht identisch. Sie können Ihre erforderliche Cluster-Speicherkapazität mit wenigen großen Knoten oder mit vielen kleineren Knoten abdecken. Mit wechselnden Anforderungen können Sie Knoten zum Cluster hinzufügen oder aus diesem entfernen und zahlen so nur für die Knoten, die Sie tatsächlich brauchen.

Die Gesamtspeicherkapazität Ihres Clusters wird berechnet, indem die Anzahl der Knoten im Cluster mit der RAM Kapazität jedes Knotens nach Abzug des System-Overheads multipliziert wird. Die Kapazität jedes Knotens basiert auf dem Knotentyp.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

Die Anzahl von Knoten im Cluster ist ein Schlüsselfaktor für die Verfügbarkeit Ihres Clusters, der Memcached ausführt. Der Ausfall eines einzelnen Knotens kann Auswirkungen auf die Verfügbarkeit Ihrer Anwendung und die Belastung Ihrer Backend-Datenbank haben. Stellt in einem solchen Fall einen Ersatz ElastiCache für einen ausgefallenen Knoten bereit und dieser wird wieder aufgefüllt. Um diese Auswirkungen auf die Verfügbarkeit zu verringern, sollten Sie die Speicher- und

Rechenkapazität auf mehrere Knoten mit geringerer Kapazität verteilen, anstatt weniger Knoten mit hoher Kapazität zu verwenden.

In einem Szenario, in dem Sie 35 GB Cache-Speicher haben möchten, können Sie eine der folgenden Konfigurationen einrichten:

- 11 `cache.t2.medium`-Knoten mit 3,22 GB Speicher und jeweils 2 Threads = 35,42 GB und 22 Threads.
- 6 `cache.m4.large`-Knoten mit 6,42 GB Speicher und jeweils 2 Threads = 38,52 GB und 12 Threads.
- 3 `cache.r4.large`-Knoten mit 12,3 GB Speicher und jeweils 2 Threads = 36,90 GB und 6 Threads.
- 3 `cache.m4.xlarge`-Knoten mit 14,28 GB Speicher und jeweils 4 Threads = 42,84 GB und 12 Threads.

Vergleichen der Knotenoptionen

Knotentyp	Speicher (in GiB)	Kerne	Kosten pro Stunde*	Erforderliche Knoten	Gesamtspeicher (in GiB)	Kerne gesamt	Monatlicher Preis
<code>cache.t2.medium</code>	3,22	2	0,068 USD	11	35,42	22	538,56 USD
<code>cache.m4.large</code>	6,42	2	0,156 USD	6	38,52	12	673,92 USD
<code>cache.m4.xlarge</code>	14,28	4	0,311 USD	3	42,84	12	671,76 USD
<code>cache.m5.xlarge</code>	12,93	4	0,311 USD	3	38,81	12	671,76 USD
<code>cache.m6g.large</code>	6,85	2	\$0,147	6	41,1	12	\$635
<code>cache.r4.large</code>	12,3	2	0,228 USD	3	36,9	6	492,48 USD

Knotentyp	Speicher (in GiB)	Kerne	Kosten pro Stunde*	Erforderliche Knoten	Gesamtspeicher (in GiB)	Kerne gesamt	Monatlicher Preis
cache.r5.large	13,07	2	0,216 USD	3	39,22	6	466,56 USD
cache.r6g.large	13,07	2	\$0,205	3	42,12	6	\$442

* Preis pro Stunde und Knoten ab 08. Oktober 2020.

Kosten pro Monat bei 100 %-iger Auslastung für 30 Tage (720 Stunden).

Diese Optionen bieten jeweils ähnliche Speicherkapazität bei unterschiedlicher Rechenkapazität und Kosten. Informationen zum Vergleich der Kosten Ihrer spezifischen Optionen finden Sie unter [ElastiCache Amazon-Preise](#).

Für Cluster, die Memcached ausführen, wird ein Teil des verfügbaren Speichers auf jedem Knoten für den Verbindungs-Overhead verwendet. Weitere Informationen finden Sie unter [Overhead von Memcached-Verbindungen](#).

Bei der Verwendung mehrerer Knotenpunkte müssen die Schlüssel auf diese verteilt werden. Jeder Knoten besitzt einen eigenen Endpunkt. Für eine einfache Endpunktverwaltung können Sie ElastiCache die Auto Discovery-Funktion verwenden, mit der Client-Programme automatisch alle Knoten in einem Cluster identifizieren können. Weitere Informationen finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#).

In manchen Fällen sind Sie vielleicht unsicher, wie viel Kapazität Sie benötigen. Wenn das so ist, empfehlen wir, zu Testzwecken mit einem cache.m5.large-Knoten zu beginnen. Überwachen Sie dann die Speichernutzung, CPU Auslastung und Cache-Trefferquote anhand der auf Amazon veröffentlichten ElastiCache Metriken CloudWatch. Weitere Informationen zu CloudWatch Metriken für ElastiCache finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#). Für die Produktion und größere Workloads bieten die R5-Knoten die beste Leistung und das beste RAM Preis-Leistungs-Verhältnis.

Wenn Ihr Cluster nicht die gewünschte Trefferquote aufweist, können Sie einfach weitere Knoten hinzufügen, um den gesamten verfügbaren Speicher in Ihrem Cluster zu erhöhen.

Wenn Ihr Cluster gebunden ist, CPU aber eine ausreichende Trefferquote aufweist, richten Sie einen neuen Cluster mit einem Knotentyp ein, der mehr Rechenleistung bietet.

Einen Cluster für Valkey oder Redis erstellen OSS

Die folgenden Beispiele zeigen, wie Sie einen Valkey- oder OSS Redis-Cluster mithilfe von, und erstellen. [AWS Management Console](#) [AWS CLI](#) [ElastiCache API](#)

Einen Valkey oder Redis erstellen OSS (Clustermodus deaktiviert) (Konsole)

ElastiCache unterstützt die Replikation, wenn Sie die Valkey- oder Redis-Engine verwenden.

OSS Um die Latenz zwischen dem Schreiben von Daten in einen primären Valkey- oder OSS Redis-Lese-Schreib-Cluster und der Weitergabe an einen schreibgeschützten sekundären Cluster zu überwachen, ElastiCache fügt dem Cluster einen speziellen Schlüssel hinzu.

`ElastiCacheMasterReplicationTimestamp` Dieser Schlüssel ist die aktuelle Weltzeit ().

UTC Da ein Valkey- oder OSS Redis-Cluster möglicherweise zu einem späteren Zeitpunkt zu einer Replikationsgruppe hinzugefügt wird, ist dieser Schlüssel in allen Valkey- oder OSS Redis-Clustern enthalten, auch wenn sie anfänglich keine Mitglieder einer Replikationsgruppe sind. Weitere Informationen zu Replikationsgruppen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Gehen Sie wie unter beschrieben vor, um einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) zu erstellen. [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Valkey- oder Redis-Clusters OSS (Cluster-Modus aktiviert) (Konsole)

Wenn Sie Redis OSS 3.2.4 oder höher ausführen, können Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) erstellen. Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) unterstützen die Partitionierung Ihrer Daten auf 1 bis 500 Shards (API/CLI: Knotengruppen), jedoch mit einigen Einschränkungen. Einen Vergleich von Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert) finden Sie unter. [Unterstützte Engines und Versionen](#)

So erstellen Sie mit der Konsole einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) ElastiCache

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der Sie diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Erstellen VPC und folgen Sie den unter [Erstellen einer virtuellen privaten Cloud \(\) VPC](#) beschriebenen Schritten.
5. Wählen Sie auf der ElastiCache Dashboard-Seite Create Cluster und dann Create Valkey cluster oder Create Redis OSS cluster aus.
6. Führen Sie unter Cluster settings (Cluster-Einstellungen) die folgenden Schritte aus:
 - a. Wählen Sie Configure and create a new cluster (Neuen Cluster konfigurieren und erstellen) aus.
 - b. Wählen Sie für Cluster mode (Cluster-Modus) Enabled (Aktiviert) aus.
 - c. Geben Sie für Cluster info (Cluster-Info) einen Wert für Name (Name) ein.
 - d. (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
7. Unter Location (Speicherort):

AWS Cloud

1. Wir empfehlen für AWS Cloud die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache \(Redis OSS\)](#) mit Multi-AZ.
2. Unter Cluster settings (Cluster-Einstellungen):
 - a. Wählen Sie unter Engine version (Engine-Version) eine verfügbare Version aus.
 - b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.
 - c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis-Parameter OSS](#) und [Eine ElastiCache Parametergruppe erstellen](#).

Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parteigruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼).

Wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

- e. Wählen Sie unter Anzahl der Shards die Anzahl der Shards (Partitionen/Knotengruppen) aus, die Sie für diesen Valkey- oder Redis-Cluster (Clustermodus aktiviert) benötigen. OSS

Bei einigen Versionen von Valkey oder Redis OSS (Clustermodus aktiviert) können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern:

- Redis OSS 3.2.10 und höher — Wenn auf Ihrem Cluster Redis OSS 3.2.10 oder spätere Versionen ausgeführt werden, können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern. Weitere Informationen finden Sie unter [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#).
 - Andere OSS Redis-Versionen — Wenn auf Ihrem Cluster eine Version von Redis vor Version 3.2.10 ausgeführt wird, gibt es einen anderen Ansatz OSS. Um die Anzahl der Shards in Ihrem Cluster in diesem Fall zu ändern, erstellen Sie einen neuen Cluster mit der neuen Anzahl von Shards. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
- f. Wählen Sie für Replicas per shard (Replikate pro Shard) die Anzahl der Read Replica-Knoten aus, die sich in jedem Shard befinden sollen.

Die folgenden Einschränkungen gelten für Valkey oder Redis OSS (Clustermodus aktiviert).

- Wenn Sie Multi-AZ aktiviert haben, stellen Sie sicher, dass mindestens ein Replikat pro Shard vorhanden ist.
- Die Anzahl der Replikate ist für jeden Shard gleich, wenn der Cluster mithilfe der Konsole erstellt wird.
- Die Anzahl der Lesereplikate pro Shard ist ein fester Wert, der nicht geändert werden kann. Wenn Sie feststellen, dass Sie mehr oder weniger Replikate pro Shard (API/CLI: Knotengruppe) benötigen, müssen Sie einen neuen Cluster mit der neuen Anzahl von Replikaten erstellen. Weitere Informationen finden Sie unter [Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup](#).

3. Unter Connectivity (Konnektivität)

- a. Wählen Sie als Network type (Netzwerktyp) die IP-Version(en) aus, die dieser Cluster unterstützen soll.
- b. Wählen Sie unter Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen. ElastiCache Cluster benötigen ein Dual-Stack-Subnetz mit beiden IPv4 und ihnen zugewiesenen IPv6 Adressen, um im Dual-Stack-Modus betrieben werden zu können, und ein Subnetz nur für den Betrieb als IPv6 -only.
IPv6

Geben Sie beim Erstellen einer neuen Subnetzgruppe die ID ein, zu der sie gehört.
VPC

Wählen Sie einen Discovery IP type (Erkennungs-IP-Typ) aus. Es werden nur die IP-Adressen des von Ihnen ausgewählten Protokolls zurückgegeben.

Weitere Informationen finden Sie unter:

- [Auswahl eines Netzwerktyps in ElastiCache](#).
- [Erstellen Sie ein Subnetz in Ihrem VPC](#).

Wenn Sie [Verwenden von lokalen Zonen mit ElastiCache](#) sind, müssen Sie ein Subnetz erstellen oder auswählen, das in der lokalen Zone liegt.

Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).

4. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:

- Keine Präferenz — ElastiCache wählt die Availability Zone.
- Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

5. Wählen Sie Next (Weiter).

6. Unter Erweiterte Valkey-Einstellungen oder Erweiterte OSS Redis-Einstellungen oder

- Für Security (Sicherheit):
 - i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:
 - Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie „Vom Kunden verwalteter AWS KMS Schlüssel“ und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Vom Kunden verwaltete Schlüssel verwenden von AWS KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter

[Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2 und höher oder Redis OSS Engine Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:

- Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
- Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Benutzergruppen mit der Konsole verwalten und CLI](#).
- AUTHStandardbenutzer — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis-Server. OSS Weitere Informationen finden Sie unter [AUTH](#).
- AUTH— Ein Authentifizierungsmechanismus für Valkey- oder Redis-ServerOSS. Weitere Informationen finden Sie unter [AUTH](#).

 Note

Für OSS Redis-Versionen zwischen 3.2.6 und höher, mit Ausnahme von Version 3.2.10, AUTH ist dies die einzige Option.

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre verwenden VPC oder eine neue erstellen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Sie VPC](#) im VPCAmazon-Benutzerhandbuch.
7. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zur Sicherung und Wiederherstellung finden Sie unter [Snapshot und Wiederherstellung](#).

- (Optional) Geben Sie ein Wartungsfenster an. Das Wartungsfenster ist der Zeitraum, der in der Regel eine Stunde lang ist, jede Woche, zu der ElastiCache die Systemwartung für Ihren Cluster geplant wird. Sie können ElastiCache den Tag und die Uhrzeit für Ihr Wartungsfenster wählen lassen (keine Präferenz), oder Sie können Tag, Uhrzeit und Dauer selbst wählen (Wartungsfenster angeben). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Zeiten sind UCT Zeiten.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

- (Optional) Für Protokolle:
 - Wählen Sie unter Protokollformat entweder Text oder aus JSON.
 - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Streamnamen aus.
- Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).
- Wählen Sie Weiter.
- Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Sobald Sie bereit sind, klicken Sie auf Create (Erstellen).

On premises

- Für On premises (On-Premises) empfehlen wir Ihnen, Auto-failover (Automatisches Failover) aktiviert zu lassen. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten in ElastiCache \(RedisOSS\)](#) mit Multi-AZ
- Befolgen Sie die Schritte unter [Verwenden von Outposts](#).

Um das Äquivalent mithilfe von ElastiCache API oder AWS CLI anstelle der ElastiCache Konsole zu erstellen, gehen Sie wie folgt vor:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Sobald der Status Ihres Clusters verfügbar ist, können Sie EC2 Zugriff darauf gewähren, eine Verbindung herstellen und ihn verwenden. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

 **Important**

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Clusters (AWS CLI)

Um einen Cluster mit dem zu erstellen AWS CLI, verwenden Sie den `create-cache-cluster` Befehl.

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) () CLI

Example — Ein Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) ohne Read Replicas

Der folgende CLI Code erstellt einen Valkey- oder Redis-Cache-Cluster OSS (Clustermodus deaktiviert) ohne Replikate.

Note

Wenn Sie einen Cluster mit einem Knotentyp aus der R6gd-Familie erstellen, müssen Sie den Parameter `data-tiering-enabled` übergeben.

Für Linux, macOS oder 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
```

Für 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
```

Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) ()AWS CLI

Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) (API/CLI: Replikationsgruppen) können mit diesem Vorgang nicht erstellt werden. `create-cache-cluster` Informationen zum Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) finden Sie unter. [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen \(\)AWS CLI](#)

Weitere Informationen finden Sie im AWS CLI ElastiCache Referenzthema. [create-replication-group](#)

Einen Cluster für Valkey oder Redis OSS erstellen () ElastiCache API

Verwenden Sie die `CreateCacheCluster` Aktion ElastiCache API, um einen Cluster mit dem zu erstellen.

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Themen

- [Erstellen eines Valkey- oder Redis-Cache-Clusters OSS \(Clustermodus deaktiviert\) \(\) ElastiCache API](#)
- [Erstellen eines Cache-Clusters in Valkey oder Redis OSS \(Clustermodus aktiviert\) \(\) ElastiCache API](#)

Erstellen eines Valkey- oder Redis-Cache-Clusters OSS (Clustermodus deaktiviert) () ElastiCache API

Der folgende Code erstellt einen Valkey- oder Redis-Cache-Cluster OSS (Clustermodus deaktiviert) (). ElastiCache API

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
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>
```

Erstellen eines Cache-Clusters in Valkey oder Redis OSS (Clustermodus aktiviert) () ElastiCache API

Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) (API/CLI: Replikationsgruppen) können mit diesem Vorgang nicht erstellt werden. `CreateCacheCluster` Informationen zum Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) finden Sie unter [Eine Replikationsgruppe in Valkey oder Redis OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen \(\) ElastiCache API](#)

Weitere Informationen finden Sie im ElastiCache API Referenzthema. [CreateReplicationGroup](#)

Einen Cluster für Memcached erstellen

Die folgenden Beispiele zeigen, wie ein Cluster mithilfe von AWS Management Console, AWS CLI und erstellt wird. ElastiCache API

Erstellen eines Memcached-Clusters (Konsole)

Wenn Sie die Memcached-Engine verwenden, unterstützt Amazon die horizontale Partitionierung Ihrer Daten über mehrere Knoten. Da Memcached Auto Discovery ermöglicht, brauchen Sie die Endpunkte eines jeden Knotens nicht zu verfolgen. Memcached verfolgt den Endpunkt jedes Knotens und aktualisiert die Endpunktliste beim Hinzufügen oder Entfernen von Knoten. Ihre Anwendung braucht nur mit dem Cluster zu kommunizieren, der als Konfigurationsendpunkt fungiert.

Befolgen Sie zum Erstellen eines Memcached-Clusters die Schritte unter [Erstellen eines Clusters](#).

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, eine Verbindung herstellen und mit der Nutzung beginnen. Weitere Informationen erhalten Sie unter [Schritt 3. Autorisieren Sie den Zugriff auf den Cluster](#) und [Schritt 4. Connect zum Knoten des Clusters her](#).

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Clusters (AWS CLI)

Verwenden Sie den `create-cache-cluster` Befehl AWS CLI, um einen Cluster mit dem zu erstellen.

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Erstellen eines Memcached-Cache-Clusters (AWS CLI)

Der folgende CLI Code erstellt einen Memcache-Cache-Cluster mit 3 Knoten.

Für Linux, macOS oder 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
```

Für 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
```

Einen Cluster für Memcached () erstellen ElastiCache API

Verwenden Sie die CreateCacheCluster Aktion ElastiCache API, um einen Cluster mit dem zu erstellen.

Important

Sobald Ihr Cluster verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, die der Cluster aktiv ist, auch wenn Sie ihn nicht aktiv nutzen. Damit Ihnen keine Kosten mehr für diesen Cluster entstehen, müssen Sie ihn löschen. Siehe [Löschen eines Clusters in ElastiCache](#).

Themen

- [Einen Memcache-Cache-Cluster erstellen \(\) ElastiCache API](#)

Einen Memcache-Cache-Cluster erstellen () ElastiCache API

Der folgende Code erstellt einen Memcache-Cluster mit 3 Knoten (). ElastiCache API

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
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>
```

Details eines ElastiCache Clusters anzeigen

Sie können detaillierte Informationen zu einem oder mehreren Clustern mithilfe der ElastiCache Konsole AWS CLI, oder anzeigen ElastiCache API.

Details eines Memcached-Clusters anzeigen (Konsole)

Sie können die Details eines Memcached-Clusters mit der ElastiCache Konsole, dem AWS CLI for ElastiCache oder dem anzeigen. ElastiCache API

Das folgende Verfahren beschreibt, wie Sie die Details eines Memcache-Clusters mithilfe der Konsole anzeigen können. ElastiCache

So zeigen Sie die Details eines Memcached-Clusters an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, für die Sie sich interessieren.
3. Wählen Sie im ElastiCache Engine-Dashboard Memcached aus. Dadurch wird eine Liste all Ihrer Cluster angezeigt, die auf der Memcached-Engine ausgeführt werden.
4. Um Details eines Clusters anzuzeigen, aktivieren Sie das Kontrollkästchen links neben den Namen des Clusters.
5. Wenn Sie Knoteninformationen anzeigen möchten, wählen Sie die Registerkarte Nodes (Knoten) aus, auf der Informationen zum Status und zum Endpunkt der Knoten angezeigt werden.
6. Wenn Sie Metriken anzeigen möchten, wählen Sie die Registerkarte Metrics (Metriken) aus, auf der die relevanten Metriken für alle Knoten im Cluster angezeigt werden. Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#)
7. Wählen Sie die Registerkarte Netzwerk und Sicherheit, um Details zur Netzwerkkonnektivität und zur Konfiguration der Subnetzgruppe sowie zur Sicherheitsgruppe des VPC Clusters anzuzeigen. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).
8. Wählen Sie die Registerkarte Maintenance (Wartung) aus, um Details zu den Wartungseinstellungen des Clusters anzuzeigen. Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).
9. Wählen Sie die Registerkarte Tags aus, um Details zu allen Tags anzuzeigen, die auf Cluster-Ressourcen angewendet wurden. Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#) .

Details zu Valkey oder Redis OSS (Clustermodus deaktiviert) anzeigen (Konsole)

Sie können die Details eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mit der ElastiCache Konsole, dem AWS CLI for ElastiCache oder dem anzeigen. ElastiCache API

Das folgende Verfahren beschreibt, wie Sie die Details eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mithilfe der Konsole anzeigen. ElastiCache

So zeigen Sie die Details eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Engine-Dashboard entweder Valkey oder Redis ausOSS, um eine Liste all Ihrer Cluster anzuzeigen, die auf dieser Engine laufen.
3. Markieren Sie das Kontrollkästchen links neben dem Cluster-Namen, um die Details zum Cluster anzuzeigen. Stellen Sie sicher, dass Sie einen Cluster auswählen, auf dem die Valkey- oder OSS Redis-Engine ausgeführt wird, und nicht Clustered Valkey oder Clustered Redis. OSS Damit werden Details des Clusters angezeigt, einschließlich des primären Endpunkts des Clusters.
4. Anzeigen der Knoteninformationen:
 - a. Wählen Sie den Cluster-Namen aus.
 - b. Wählen Sie die Registerkarte Shards and Nodes (Shards und Knoten) aus. Es werden die Details zu jedem Knoten angezeigt, einschließlich des Knotenendpunkts, der zum Lesen vom Cluster benötigt wird.
5. Wenn Sie Metriken anzeigen möchten, wählen Sie die Registerkarte Metrics (Metriken) aus, auf der die relevanten Metriken für alle Knoten im Cluster angezeigt werden. Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#)
6. Wählen Sie zum Anzeigen von Protokollen die Registerkarte Logs (Protokolle) aus. Sie gibt an, ob der Cluster Slow-Protokolle oder Engine-Protokolle verwendet, und enthält relevante Details. Weitere Informationen finden Sie unter [Protokollzustellung](#).
7. Wählen Sie die Registerkarte Network and security (Netzwerk und Sicherheit) aus, um Details zur Konfiguration von Netzwerkkonnektivität des Clusters und zur Subnetzgruppe anzuzeigen. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).
8. Wählen Sie die Registerkarte Maintenance (Wartung) aus, um Details zu den Wartungseinstellungen des Clusters anzuzeigen. Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

9. Wählen Sie die Registerkarte Service updates (Service-Updates) aus, um Details zu allen verfügbaren Service-Updates sowie deren empfohlenes Mindesthaltbarkeitsdatum einzusehen. Weitere Informationen finden Sie unter [Service-Updates in ElastiCache](#).
10. Wählen Sie die Registerkarte Tags aus, um Details zu allen Tags anzuzeigen, die auf Cluster-Ressourcen angewendet wurden. Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).

Details für einen Valkey- oder Redis-Cluster (Clustermodus aktiviertOSS) (Konsole) anzeigen

Sie können die Details eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) mithilfe der ElastiCache Konsole, der AWS CLI for ElastiCache oder der anzeigen. ElastiCache API

Das folgende Verfahren beschreibt, wie Sie die Details eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) mithilfe der Konsole anzeigen. ElastiCache

So zeigen Sie die Details eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, für die Sie sich interessieren.
3. Wählen Sie im ElastiCache Engine-Dashboard Valkey oder Redis aus, OSS um eine Liste all Ihrer Cluster anzuzeigen, die auf dieser Engine laufen.
4. Um Details zu einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) anzuzeigen, wählen Sie das Feld links neben dem Namen des Clusters aus. Stellen Sie sicher, dass Sie einen Cluster auswählen, auf dem die Valkey- oder Clustered Redis-Engine ausgeführt wird. OSS

Der Bildschirm wird unterhalb des Clusters erweitert und zeigt Details zum Cluster an, darunter auch den Konfigurationsendpunkt des Clusters.

5. Wenn Sie eine Auflistung der Shards des Clusters und der Anzahl der Knoten in jedem Shard anzeigen möchten, wählen Sie die Registerkarte Shards and nodes (Shards und Knoten) aus.
6. So zeigen Sie spezifische Informationen zu einem Knoten an:
 - Wählen Sie die ID des Shards aus.

Damit werden Informationen zu jedem Knoten angezeigt, einschließlich des Endpunkts des Knotens, den Sie zum Lesen von Daten aus dem Cluster benötigen.

7. Wenn Sie Metriken anzeigen möchten, wählen Sie die Registerkarte Metrics (Metriken) aus, auf der die relevanten Metriken für alle Knoten im Cluster angezeigt werden. Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#)
8. Wählen Sie zum Anzeigen von Protokollen die Registerkarte Logs (Protokolle) aus. Sie gibt an, ob der Cluster Slow-Protokolle oder Engine-Protokolle verwendet, und enthält relevante Details. Weitere Informationen finden Sie unter [Protokollzustellung](#).
9. Wählen Sie die Registerkarte Netzwerk und Sicherheit, um Details zur Netzwerkkonnektivität und Subnetzgruppenkonfiguration des Clusters, zur VPC Sicherheitsgruppe und gegebenenfalls zu der auf dem Cluster aktivierten Verschlüsselungsmethode anzuzeigen. Weitere Informationen erhalten Sie unter [Subnetze und Subnetzgruppen](#) und [Datensicherheit bei Amazon ElastiCache](#).
10. Wählen Sie die Registerkarte Maintenance (Wartung) aus, um Details zu den Wartungseinstellungen des Clusters anzuzeigen. Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).
11. Wählen Sie die Registerkarte Service updates (Service-Updates) aus, um Details zu allen verfügbaren Service-Updates sowie deren empfohlenes Mindesthaltbarkeitsdatum einzusehen. Weitere Informationen finden Sie unter [Service-Updates in ElastiCache](#).
12. Wählen Sie die Registerkarte Tags aus, um Details zu allen Tags anzuzeigen, die auf Cluster-Ressourcen angewendet wurden. Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).

Details eines ElastiCache Clusters anzeigen (AWS CLI)

Der folgende Code listet die Details für *my-cluster*:

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

Ersetzen *my-cluster* mit dem Namen Ihres Clusters für den Fall, dass der Cluster mit einem Cache-Knoten und 0 Shards mithilfe des `create-cache-cluster` Befehls erstellt wird.

```
{
  "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
}
]

```

```
}
```

In einem Fall, in dem der Cluster mithilfe von AWS Management Console (aktivierter oder deaktivierter Clusterknoten mit einem oder mehreren Shards) erstellt wird, verwenden Sie den folgenden Befehl, um die Clusterdetails zu beschreiben (ersetzen *my-cluster* mit dem Namen der Replikationsgruppe (Name Ihres Clusters)):

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Weitere Informationen finden Sie im ElastiCache Thema AWS CLI für [describe-cache-clusters](#).

Details eines ElastiCache Clusters anzeigen (ElastiCache API)

Mithilfe der ElastiCache API DescribeCacheClusters Aktion können Sie die Details für einen Cluster anzeigen. Wenn der Parameter CacheClusterId eingeschlossen wird, werden Details für den angegebenen Cluster zurückgegeben. Wenn der Parameter CacheClusterId weggelassen wird, werden Details für bis zu MaxRecords (Standard 100) Cluster zurückgegeben. Der Wert für MaxRecords darf nicht kleiner als 20 oder größer als 100 sein.

Der folgende Code listet die Details für my-cluster auf.

```
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>
```

Der folgende Code listet die Details für bis zu 25 Cluster auf.

```
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>
```

Weitere Informationen finden Sie im ElastiCache API Referenzthema [DescribeCacheClusters](#).

Einen ElastiCache Cluster ändern

Neben dem Hinzufügen oder Entfernen von Knoten aus einem ElastiCache Cluster kann es vorkommen, dass Sie weitere Änderungen vornehmen müssen, z. B. das Hinzufügen einer Sicherheitsgruppe, das Ändern des Wartungsfensters oder einer Parametergruppe.

Es wird empfohlen, dass das Wartungsfenster in den Zeitraum mit der geringsten Nutzung fällt. Dies muss folglich zeitweise korrigiert werden.

Wenn Sie die Parameter eines Clusters ändern, wird die Änderung entweder sofort oder nach dem Neustart des Clusters angewendet. Dies gilt unabhängig davon, ob Sie die Parametergruppe des Clusters oder einen Parameterwert innerhalb der Parametergruppe des Clusters ändern. Informationen darüber, wann eine bestimmte Parameteränderung angewendet wird, finden Sie im Abschnitt „Änderungen werden wirksam“ in der Spalte „Details“ in den Tabellen für [Memcached-spezifische Parameter](#) und [Valkey- und Redis-Parameter OSS](#). Weitere Informationen zum Neustarten eines Cluster-Knotens finden Sie unter [Knoten neu starten](#).

Unter Verwendung der ElastiCache AWS Management Console

So modifizieren Sie einen Cluster:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der sich der Cluster befindet, den Sie ändern möchten.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, den Sie modifizieren möchten.

Es wird eine Liste der ausgewählten Engine-Cluster angezeigt.

4. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, den Sie modifizieren möchten..
5. Wählen Sie Aktionen und dann Ändern.

Das Fenster Modify Cluster wird angezeigt.

6. Nehmen Sie im Fenster Modify Cluster die gewünschten Änderungen vor. Zu den Optionen gehören:

- Beschreibung

- Clustermodus – Wenn Sie den Clustermodus von Deaktiviert auf Aktiviert ändern möchten, müssen Sie den Clustermodus zuerst auf Kompatibel festlegen.

Im kompatiblen Modus können Ihre Valkey- oder OSS Redis-Clients sowohl im Clustermodus als auch im deaktivierten Clustermodus eine Verbindung herstellen. Nachdem Sie alle Valkey- oder OSS Redis-Clients zur Verwendung des aktivierten Clustermodus migriert haben, können Sie die Konfiguration des Clustermodus abschließen und den Clustermodus auf Aktiviert setzen.

- Kompatibilität der Engine-Version

Important

Sie können auf neuere Engine-Versionen aktualisieren. Falls Sie eine Hauptversion der Engine aktualisieren, z. B. von 5.0.6 auf 6.0, müssen Sie eine Parametergruppenfamilie auswählen, die mit der neuen Engine-Version kompatibel ist. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

- VPCSicherheitsgruppe (n)
- Parametergruppe
- Node Type

Note

Wenn der Cluster einen Knotentyp aus der R6gd-Familie verwendet, können Sie nur eine andere Knotengröße aus dieser Familie auswählen. Wenn Sie einen Knotentyp aus der R6gd-Familie wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Daten-Tiering](#).

- Multi-AZ
- Automatisches Failover (nur im deaktivierten Cluster-Modus)
- Aktivieren automatischer Sicherungen
- ID des Backup-Knotens
- Aufbewahrungszeitraum für Backups
- Backup-Fenster

- Thema für die SNS Benachrichtigung
- Versionskompatibilität der Memcached Engine
- Netzwerktyp

 Note

Wenn Sie von IPv4 zu wechseln IPv6, müssen Sie Subnetzgruppen auswählen oder erstellen, die kompatibel sind mit IPv6. Weitere Informationen finden Sie unter [Auswahl eines Netzwerktyps in ElastiCache](#).

- VPC Sicherheitsgruppe (n)
- Parametergruppe
- Wartungsfenster
- Thema für die SNS Benachrichtigung

Das Kontrollkästchen Apply Immediately (Sofort anwenden) gilt nur für Änderungen an der Engine-Version. Um die Änderungen sofort zu übernehmen, aktivieren Sie das Kontrollkästchen Apply Immediately (Sofort anwenden). Wenn dieses Kontrollkästchen nicht aktiviert ist, werden Änderungen an der Engine-Version während des nächsten Wartungsfensters übernommen. Andere Änderungen, wie z. B. Ändern des Wartungsfensters, werden umgehend übernommen.

Um die Protokollzustellung für Redis zu aktivieren/deaktivieren

1. Suchen Sie in der Liste der Cluster den Cluster aus, den Sie ändern möchten. Wählen Sie den Clusternamen und nicht das Kontrollkästchen daneben.
2. Wählen Sie auf der Seite mit den Cluster-Details die Registerkarte Logs aus.
3. Um langsame Logs zu aktivieren oder zu deaktivieren, wählen Sie entweder Aktivieren oder Deaktivieren.

Wenn Sie „Aktivieren“ wählen:

- a. Wählen Sie unter Protokollformat entweder JSON oder Text aus.
- b. Wählen Sie unter Protokollzieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.

- c. Unter Protokollziel können Sie `Create new` auswählen und entweder Ihren CloudWatchLogs Log-Gruppennamen oder Ihren Kinesis Data Firehose-Stream-Namen eingeben. Sie können auch `Select existing` auswählen und dann entweder Ihren CloudWatchLogs Loggruppenamen oder Ihren Kinesis Data Firehose Firehose-Streamnamen wählen.
- d. Wählen Sie `Enable (Aktivieren)` aus.

So ändern Sie Ihre Konfiguration für Redis:

1. Wählen Sie `Ändern` aus.
2. Wählen Sie unter Protokollformat entweder `JSON` oder `Text` aus.
3. Wählen Sie unter Zieltyp entweder `CloudWatch Logs` oder `Kinesis Firehose` aus.
4. Wählen Sie unter Protokollziel entweder `Neu erstellen` aus und geben Sie Ihren CloudWatchLogs Protokollgruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen ein. Oder wählen Sie `Select existing` und wählen Sie dann Ihren CloudWatchLogs Log-Gruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen.

Verwenden Sie das mit AWS CLI ElastiCache

Sie können einen vorhandenen Cluster mithilfe der AWS CLI `modify-cache-cluster` Operation ändern. Um den Konfigurationswert eines Clusters zu ändern, geben Sie die ID des Clusters, den zu ändernden Parameter und den neuen Wert des Parameters ein. Das folgende Beispiel ändert das Wartungsfenster für einen Cluster namens `my-cluster` und übernimmt die Änderung umgehend.

 **Important**

Sie können ein Upgrade auf neuere Versionen der Memcached Engine durchführen. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

 **Important**

Sie können auf neuere Valkey- oder OSS Redis-Engine-Versionen aktualisieren. Wenn Sie wichtige Engine-Versionen aktualisieren, z. B. von Redis OSS 5.0.6 auf Redis OSS 6.0, müssen Sie eine Parametergruppenfamilie auswählen, die mit der neuen Engine-

Version kompatibel ist. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --preferred-maintenance-window sun:23:00-mon:02:00
```

Der Parameter `--apply-immediately` gilt nur für Änderungen am Knotentyp, an der Engine-Version und der Anzahl von Knoten in einem Cluster. Wenn Sie solche Änderungen umgehend übernehmen möchten, verwenden Sie den Parameter `--apply-immediately`. Wenn Sie solche Änderungen lieber bis zum nächsten Wartungsfenster aufschieben möchten, verwenden Sie den Parameter `--no-apply-immediately`. Andere Änderungen, wie z. B. Ändern des Wartungsfensters, werden umgehend übernommen.

Weitere Informationen finden Sie im For-Thema. AWS CLI ElastiCache [modify-cache-cluster](#)

Mit dem ElastiCache API

Sie können einen vorhandenen Cluster mithilfe der ElastiCache API `ModifyCacheCluster` Operation ändern. Um den Konfigurationswert eines Clusters zu ändern, geben Sie die ID des Clusters, den zu ändernden Parameter und den neuen Wert des Parameters ein. Das folgende Beispiel ändert das Wartungsfenster für einen Cluster namens `my-cluster` und übernimmt die Änderung umgehend.

Important

Sie können ein Upgrade auf neuere Versionen der Memcached Engine durchführen. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#).

Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

⚠ Important

Sie können auf neuere Valkey- oder OSS Redis-Engine-Versionen aktualisieren. Wenn Sie wichtige Engine-Versionen aktualisieren, z. B. von Redis OSS 5.0.6 auf Redis OSS 6.0, müssen Sie eine Parametergruppenfamilie auswählen, die mit der neuen Engine-Version kompatibel ist. Weitere Informationen zur Verfahrensweise finden Sie unter [Versionsverwaltung für ElastiCache](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
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>
```

Der Parameter `ApplyImmediately` gilt nur für Änderungen am Knotentyp, an der Engine-Version und der Anzahl von Knoten in einem Cluster. Wenn Sie solche Änderungen umgehend übernehmen möchten, stellen Sie den Parameter `ApplyImmediately` auf `true` ein. Wenn Sie solche Änderungen lieber bis zum nächsten Wartungsfenster aufschieben möchten, stellen Sie den Parameter `ApplyImmediately` auf `false` ein. Andere Änderungen, wie z. B. Ändern des Wartungsfensters, werden umgehend übernommen.

Weitere Informationen finden Sie im Referenzthema. ElastiCache API [ModifyCacheCluster](#)

Hinzufügen von Knoten zu einem ElastiCache Cluster

Das Hinzufügen von Knoten zu einem Memcached-Cluster erhöht die Anzahl Ihrer Cluster-Partitionen. Wenn Sie die Anzahl der Partitionen in einem Cluster ändern, müssen einige Ihrer Schlüsselbereiche neu zugeordnet werden, damit sie auf den richtigen Knoten abgebildet werden. Die Neuordnung von Schlüsselräumen erhöht vorübergehend die Anzahl der Cache-Fehler auf dem Cluster. Weitere Informationen finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

Informationen zur Neukonfiguration Ihres Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) finden Sie unter [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)

Sie können die ElastiCache Management Console, das AWS CLI oder verwenden, ElastiCache API um Ihrem Cluster Knoten hinzuzufügen.

Mit dem ElastiCache AWS Management Console

Wenn Sie einen Knoten zu einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem einzigen Knoten (ohne aktivierte Replikation) hinzufügen möchten, müssen Sie zuerst die Replikation hinzufügen und dann einen Replikatknoten hinzufügen.

Themen

- [Um die Replikation zu einem Valkey- oder OSS Redis-Cluster ohne Shards hinzuzufügen](#)
- [Um Knoten zu einem ElastiCache Cluster hinzuzufügen \(Konsole\)](#)

Das folgende Verfahren fügt die Replikation zu einem Valkey- oder OSS Redis-Einzelknoten hinzu, für den die Replikation nicht aktiviert ist. Wenn Sie Replikation hinzufügen, wird der vorhandene Knoten zum primären Knoten im replikationsfähigen Cluster. Nachdem die Replikation hinzugefügt wurde, können Sie dem Cluster bis zu 5 Replikationsknoten hinzufügen.

Um die Replikation zu einem Valkey- oder OSS Redis-Cluster ohne Shards hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.

Eine Liste der Cluster, auf denen diese Engine ausgeführt wird, wird angezeigt.

3. Wählen Sie den Namen eines Clusters, nicht das Kästchen links vom Namen des Clusters, dem Sie Knoten hinzufügen möchten.

Folgendes gilt für einen OSS Redis-Cluster, für den die Replikation nicht aktiviert ist:

- Es läuft RedisOSS, nicht Clustered Redis. OSS
- Er besitzt keine Shards.

Wenn der Cluster über Shards verfügt, ist die Replikation für ihn bereits aktiviert und Sie können mit [Um Knoten zu einem ElastiCache Cluster hinzuzufügen \(Konsole\)](#) fortfahren.

4. Wählen Sie Add replication.
5. Geben Sie unter Add Replication eine Beschreibung für diesen replikationsfähigen Cluster ein.
6. Wählen Sie Hinzufügen aus.

Sobald der Status des Clusters wieder available lautet, können Sie mit dem nächsten Verfahren fortfahren und Replikate zum Cluster hinzufügen.

Um Knoten zu einem ElastiCache Cluster hinzuzufügen (Konsole)

Die folgende Vorgehensweise kann verwendet werden, um Knoten zu einem Cluster hinzuzufügen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, dem Sie Knoten hinzufügen möchten.

Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.

3. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, dem Sie einen Knoten hinzufügen möchten..

Wenn es sich bei Ihrem Cluster um einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) handelt, finden Sie weitere Informationen unter. [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)

Wenn es sich bei Ihrem Cluster um einen Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) ohne Shards handelt, führen Sie zunächst die Schritte unter aus. [Um die Replikation zu einem Valkey- oder OSS Redis-Cluster ohne Shards hinzuzufügen](#)

4. Wählen Sie Knoten hinzufügen aus.

5. Geben Sie im Dialogfeld Add Node (Knoten hinzufügen) die erforderlichen Informationen ein.
6. Wählen Sie die Schaltfläche Sofort übernehmen – Ja, um diesen Knoten sofort hinzuzufügen, oder Nein, um das Hinzufügen dieses Knotens bis zum nächsten Wartungsfenster des Clusters aufzuschieben.

Auswirkung von neuen Hinzufügungs- und Entfernungsanträgen auf ausstehende Anträge

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
Szenario 1	Löschen	Löschen	<p>Der neue Löschantrag, ob anstehend oder sofort, ersetzt den anstehenden Löschantrag..</p> <p>Wenn z. B. die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Löschantrag für die Knoten 0002 und 0004 gestellt wird, werden nur die Knoten 0002 und 0004 gelöscht. Die Knoten 0001, 0003 und 0007 werden nicht gelöscht.</p>
Szenario 2	Löschen	Erstellen	<p>Der neue Erstellungsauftrag, ob anstehend oder sofort, ersetzt den ausstehenden Löschauftrag.</p> <p>Wenn beispielsweise die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Antrag auf Erstellung eines Knotens gestellt wird, wird ein neuer Knoten erstellt und die Knoten 0001, 0003 und 0007 werden nicht gelöscht.</p>
Szenario 3	Erstellen	Löschen	<p>Der neue Löschantrag, ob anstehend oder sofort, ersetzt den ausstehenden Erstellungsantrag.</p> <p>Sind beispielsweise zwei Knoten anzulegen und wird ein neuer Antrag auf Löschung des Knotens 0003 gestellt, werden keine neuen Knoten angelegt und der Knoten 0003 wird gelöscht.</p>
Szenario 4	Erstellen	Erstellen	Der neue Erstellungsauftrag wird dem ausstehenden Erstellungsauftrag hinzugefügt.

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
			<p>Wenn z. B. eine ausstehende Anforderung zur Erstellung von zwei Knoten besteht und eine neue Anforderung zur Erstellung von drei Knoten ausgegeben wird, wird die neue Anforderung der ausstehenden Anforderung hinzugefügt und es werden fünf Knoten erstellt.</p> <div data-bbox="764 625 1507 1031" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9e6;"> <p> Important</p> <p>Wenn der neue Erstellungsauftrag auf Sofort anwenden – Ja eingestellt ist, werden alle Erstellungsaufträge sofort ausgeführt. Wenn der neue Erstellungsauftrag auf Sofort anwenden – Nein eingestellt ist, sind alle Erstellungsaufträge in der Schwebe.</p> </div>

Um festzustellen, welche Vorgänge noch nicht abgeschlossen sind, wählen Sie die Registerkarte Beschreibung und prüfen Sie, wie viele ausstehende Erstellungen oder Löschungen angezeigt werden. Sie können nicht sowohl ausstehende Erstellungen als auch ausstehende Löschungen haben.

7. Wählen Sie die Schaltfläche Add.

Nach kurzer Zeit sollten die neuen Knoten mit dem Status `creating` in der Knotenliste erscheinen. Wenn dies nicht der Fall ist, aktualisieren Sie die Browserseite. Sobald sich der Status des Knotens in `verfügbar` ändert, kann der neue Knoten verwendet werden.

Verwenden Sie den AWS CLI ElastiCache

Um einem Cluster mithilfe von Knoten hinzuzufügen AWS CLI, verwenden Sie den AWS CLI Vorgang `modify-cache-cluster` mit den folgenden Parametern:

- `--cache-cluster-id` Die ID des Cache-Clusters, zu dem Sie Knoten hinzufügen möchten.

- `--num-cache-nodes` Der Parameter `--num-cache-nodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `--num-cache-nodes` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein. Wenn dieser Wert kleiner als die aktuelle Anzahl von Knoten ist, ElastiCache erwartet, dass der Parameter `cache-node-ids-to-remove` und eine Liste von Knoten aus dem Cluster entfernt werden. Weitere Informationen finden Sie unter [Verwenden Sie mit AWS CLI ElastiCache](#).
- `--apply-immediately` oder `--no-apply-immediately` gibt an, ob diese Knoten umgehend bzw. während des nächsten Wartungsfensters hinzugefügt werden sollen.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 5 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 5 ^  
  --apply-immediately
```

Dieser Vorgang erzeugt eine Ausgabe, die der folgenden ähnelt (JSONFormat):

```
{  
  "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",
  }
}
```

Weitere Informationen finden Sie im AWS CLI Thema [modify-cache-cluster](#).

Verwenden von AWS CLI with ElastiCache

Wenn Sie Knoten zu einem vorhandenen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) hinzufügen möchten, für den die Replikation nicht aktiviert ist, müssen Sie zuerst die Replikationsgruppe erstellen und dabei den vorhandenen Cluster als primären Cluster angeben. Weitere Informationen finden Sie unter [Erstellen einer Replikationsgruppe mithilfe eines verfügbaren Valkey- oder OSS Redis-Cache-Clusters \(AWS CLI\)](#). Wenn die Replikationsgruppe verfügbar ist, können Sie mit dem folgenden Vorgang fortfahren.

Um einem Cluster mithilfe von Knoten hinzuzufügen AWS CLI, verwenden Sie den AWS CLI Vorgang `increase-replica-count` mit den folgenden Parametern:

- `--replication-group-id` Die ID der Replikationsgruppe, aus der Sie Knoten entfernen möchten.

- `--new-replica-count` gibt die Anzahl der Knoten an, die nach Anwendung der Änderung in dieser Replikationsgruppe enthalten sein sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `--new-replica-count` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein.
- `--apply-immediately` oder `--no-apply-immediately` gibt an, ob diese Knoten umgehend bzw. während des nächsten Wartungsfensters hinzugefügt werden sollen.

Für Linux, macOS oder Unix:

```
aws elasticache increase-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 4 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache increase-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 4 ^  
  --apply-immediately
```

Dieser Vorgang erzeugt eine Ausgabe, die der folgenden ähnelt (JSONFormat):

```
{  
  "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"
  }
}
```

Weitere Informationen finden Sie im AWS CLI Thema [increase-replica-count](#).

Mit dem ElastiCache API

Wenn Sie Knoten zu einem vorhandenen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) hinzufügen möchten, für den die Replikation nicht aktiviert ist, müssen Sie zuerst die Replikationsgruppe erstellen und den vorhandenen Cluster als primären Cluster angeben. Weitere Informationen finden Sie unter [Hinzufügen von Replikaten zu einem eigenständigen Valkey- oder Redis-Cluster OSS \(Clustermodus deaktiviert\) \(\) ElastiCache API](#). Wenn die Replikationsgruppe verfügbar ist, können Sie mit dem folgenden Vorgang fortfahren.

Um Knoten zu einem Cluster hinzuzufügen () ElastiCache API

- Rufen Sie den `IncreaseReplicaCount` API Vorgang mit den folgenden Parametern auf:
 - `ReplicationGroupId` Die ID des Clusters, zu dem Sie Knoten hinzufügen möchten.
 - `NewReplicaCount` Der Parameter `NewReplicaCount` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `NewReplicaCount` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein. Wenn dieser Wert unter der aktuellen Anzahl von Knoten liegt, verwenden Sie den `DecreaseReplicaCount` API zusammen mit der Anzahl der Knoten, die aus dem Cluster entfernt werden sollen.
 - `ApplyImmediately` Gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters hinzugefügt werden sollen.
 - `Region` Gibt die AWS Region des Clusters an, zu der Sie Knoten hinzufügen möchten.

Das folgende Beispiel zeigt einen Aufruf zum Hinzufügen von Knoten zu einem 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>
```

Weitere Informationen finden Sie unter ElastiCache API Thema [IncreaseReplicaCount](#).

Mit dem ElastiCache API

Um Knoten zu einem Cluster hinzuzufügen (ElastiCache API)

- Rufen Sie den `ModifyCacheCluster` API Vorgang mit den folgenden Parametern auf:
 - `CacheClusterId` Die ID des Clusters, zu dem Sie Knoten hinzufügen möchten.
 - `NumCacheNodes` Der Parameter `NumCachNodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `NumCacheNodes` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein. Wenn dieser Wert unter der aktuellen Anzahl von Knoten liegt, wird ElastiCache erwartet, dass der Parameter `CacheNodeIdsToRemove` mit einer Liste von Knoten aus dem Cluster entfernt wird (siehe [Verwenden von ElastiCache API mit Memcached](#)).
 - `ApplyImmediately` Gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters hinzugefügt werden sollen.

- `Region` Gibt die AWS Region des Clusters an, zu der Sie Knoten hinzufügen möchten.

Das folgende Beispiel zeigt einen Aufruf zum Hinzufügen von Knoten zu einem 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>
```

Weitere Informationen finden Sie unter ElastiCache API Thema [ModifyCacheCluster](#).

Knoten aus einem ElastiCache Cluster entfernen

Sie können einen Knoten mit dem, dem oder dem aus einem Valkey-OSS, Redis- oder Memcached-Cluster löschen. AWS Management Console AWS CLI ElastiCache API

Note

Jedes Mal, wenn Sie die Anzahl der Knoten in einem Memcached-Cluster ändern, müssen Sie zumindest einen Teil Ihres Keyspace neu zuordnen, sodass er dem richtigen Knoten zugeordnet ist. Detaillierte Informationen zum Lastausgleich eines Memcached-Clusters finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

Mit dem ElastiCache AWS Management Console

Entfernen von Knoten aus einem Cluster (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region des Clusters aus, aus dem Sie Knoten entfernen möchten.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, von dem Sie einen Knoten entfernen möchten.

Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.

4. Wählen Sie aus der Liste der Cluster den Namen des Clusters aus, aus dem Sie einen Knoten entfernen möchten.

Eine Liste der Knoten des Clusters wird angezeigt.

5. Aktivieren Sie das Kontrollkästchen links neben der Knoten-ID des Knotens, den Sie entfernen möchten. Mit der ElastiCache Konsole können Sie jeweils nur einen Knoten löschen. Wenn Sie also mehrere Knoten auswählen, können Sie die Schaltfläche Knoten löschen nicht verwenden.

Die Seite Delete Node (Knoten löschen) wird angezeigt.

6. Um den Knoten zu löschen, machen Sie alle Angaben auf der Seite Delete Node und wählen Sie Delete Node. Um den Knoten beizubehalten, wählen Sie Cancel (Abbrechen).

⚠ Important

Wenn Sie mit Valkey oder Redis die Knotenergebnisse im Cluster löschenOSS, die nicht mehr Multi-AZ-konform sind, stellen Sie sicher, dass Sie zuerst das Multi-AZ-Kontrollkästchen deaktivieren und dann den Knoten löschen. Wenn Sie das Kontrollkästchen Multi-AZ deaktivieren, können Sie Auto failover (Automatisches Failover) aktivieren.

Auswirkung von neuen Hinzufügungs- und Entfernungsanträgen auf ausstehende Anträge

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
Szenario 1	Löschen	Löschen	<p>Der neue Löschantrag, ob anstehend oder sofort, ersetzt den anstehenden Löschantrag..</p> <p>Wenn z. B. die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Löschantrag für die Knoten 0002 und 0004 gestellt wird, werden nur die Knoten 0002 und 0004 gelöscht. Die Knoten 0001, 0003 und 0007 werden nicht gelöscht.</p>
Szenario 2	Löschen	Erstellen	<p>Der neue Erstellungsauftrag, ob anstehend oder sofort, ersetzt den ausstehenden Löschauftrag.</p> <p>Wenn beispielsweise die Knoten 0001, 0003 und 0007 zur Löschung anstehen und ein neuer Antrag auf Erstellung eines Knotens gestellt wird, wird ein neuer Knoten erstellt und die Knoten 0001, 0003 und 0007 werden nicht gelöscht.</p>
Szenario 3	Erstellen	Löschen	<p>Der neue Löschantrag, ob anstehend oder sofort, ersetzt den ausstehenden Erstellungsantrag.</p> <p>Sind beispielsweise zwei Knoten anzulegen und wird ein neuer Antrag auf Löschung des Knotens 0003</p>

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
			gestellt, werden keine neuen Knoten angelegt und der Knoten 0003 wird gelöscht.
Szenario 4	Erstellen	Erstellen	<p>Der neue Erstellungsauftrag wird dem ausstehenden Erstellungsauftrag hinzugefügt.</p> <p>Wenn z. B. eine ausstehende Anforderung zur Erstellung von zwei Knoten besteht und eine neue Anforderung zur Erstellung von drei Knoten ausgegeben wird, wird die neue Anforderung der ausstehenden Anforderung hinzugefügt und es werden fünf Knoten erstellt.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Wenn der neue Erstellungsauftrag auf Sofort anwenden – Ja eingestellt ist, werden alle Erstellungsaufträge sofort ausgeführt. Wenn der neue Erstellungsauftrag auf Sofort anwenden – Nein eingestellt ist, sind alle Erstellungsaufträge in der Schwebe.</p> </div>

Um festzustellen, welche Vorgänge noch nicht abgeschlossen sind, wählen Sie die Registerkarte Beschreibung und prüfen Sie, wie viele ausstehende Erstellungen oder Löschungen angezeigt werden. Sie können nicht sowohl ausstehende Erstellungen als auch ausstehende Löschungen haben.

Verwenden Sie mit AWS CLI ElastiCache

1. Identifizieren Sie IDs den Knoten, den Sie entfernen möchten. Weitere Informationen finden Sie unter [Details eines ElastiCache Clusters anzeigen](#).
2. Verwenden Sie den `decrease-replica-count` CLI Vorgang mit einer Liste der zu entfernenden Knoten, wie im folgenden Beispiel.

Um über die Befehlszeilenschnittstelle Knoten aus einem Cluster zu entfernen, verwenden Sie den Befehl `decrease-replica-count` mit den folgenden Parametern:

- `--replication-group-id` Die ID der Replikationsgruppe, aus der Sie Knoten entfernen möchten.
- `--new-replica-count` Der Parameter `--new-replica-count` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- `--replicas-to-remove` Eine Liste der KnotenIDs, die Sie aus diesem Cluster entfernen möchten.
- `--apply-immediately` oder `--no-apply-immediately` gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters entfernt werden sollen.
- `--region` Gibt die AWS Region des Clusters an, aus dem Sie Knoten entfernen möchten.

Note

Sie können nur einen von `--replicas-to-remove` oder `--new-replica-count` Parametern übergeben, wenn Sie diesen Vorgang aufrufen.

Für Linux, macOS oder Unix:

```
aws elasticache decrease-replica-count \  
  --replication-group-id my-replication-group \  
  --new-replica-count 2 \  
  --region us-east-2 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache decrease-replica-count ^  
  --replication-group-id my-replication-group ^  
  --new-replica-count 3 ^  
  --region us-east-2 ^  
  --apply-immediately
```

Dieser Vorgang erzeugt eine Ausgabe, die der folgenden ähnelt (JSONFormat):

```

{
  "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"
}
}

```

Alternativ können Sie auch `decrease-replica-count` aufrufen und statt des Parameters `--new-replica-count` den Parameter `--replicas-to-remove` übergeben, wie im Folgenden gezeigt:

Für Linux, macOS oder Unix:

```

aws elasticache decrease-replica-count \
  --replication-group-id my-replication-group \
  --replicas-to-remove node-test-003 \
  --region us-east-2 \
  --apply-immediately

```

Für Windows:

```

aws elasticache decrease-replica-count ^
  --replication-group-id my-replication-group ^
  --replicas-to-remove node-test-003 ^

```

```
--region us-east-2 ^  
--apply-immediately
```

Weitere Informationen finden Sie in den AWS CLI Themen [decrease-replica-count](#).

Verwenden von ElastiCache API mit Valkey oder Redis OSS

Um Knoten mit dem zu entfernen ElastiCache API, rufen Sie den DecreaseReplicaCount API Vorgang mit der Replikationsgruppen-ID und einer Liste der zu entfernenden Knoten auf, wie hier gezeigt:

- **ReplicationGroupId** Die ID der Replikationsgruppe, aus der Sie Knoten entfernen möchten.
- **ReplicasToRemove** Der Parameter **ReplicasToRemove** gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- **ApplyImmediately** gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters entfernt werden sollen.
- **Region** Gibt die AWS Region des Clusters an, aus der Sie einen Knoten entfernen möchten.

Das folgende Beispiel entfernt umgehend Knoten 0004 und 0005 aus dem 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>
```

Weitere Informationen finden Sie unter ElastiCache API Thema [DecreaseReplicaCount](#).

Verwenden von ElastiCache API mit Memcached

Um Knoten mithilfe von zu entfernen ElastiCache API, rufen Sie den `ModifyCacheCluster` API Vorgang mit der Cache-Cluster-ID und einer Liste der zu entfernenden Knoten auf, wie hier gezeigt:

- `CacheClusterId` Die ID des Cache-Clusters, aus dem Sie die Knoten entfernen möchten.
- `NumCacheNodes` Der Parameter `NumCacheNodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- `CacheNodeIdsToRemove.member.n` Die Liste der KnotenIDs, die aus dem Cluster entfernt werden sollen.
 - `CacheNodeIdsToRemove.member.1=0004`
 - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters entfernt werden sollen.
- `Region` Gibt die AWS Region des Clusters an, aus dem Sie einen Knoten entfernen möchten.

Das folgende Beispiel entfernt umgehend Knoten 0004 und 0005 aus dem 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>
```

Weitere Informationen finden Sie unter ElastiCache API Thema [ModifyCacheCluster](#).

Abbrechen ausstehender Operationen zum Hinzufügen oder Löschen von Knoten in ElastiCache

Wenn Sie sich dafür entschieden haben, eine ElastiCache Clusteränderung nicht sofort anzuwenden, hat der Vorgang den Status Ausstehend, bis er in Ihrem nächsten Wartungsfenster ausgeführt wird. Anstehende Operationen können storniert werden.

So stornieren Sie eine anstehende Operation

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der Sie einen ausstehenden Vorgang zum Hinzufügen oder Löschen von Knoten abbrechen möchten.
3. Wählen Sie im Navigationsfeld die Engine aus, die auf dem Cluster ausgeführt wird, auf dem sich anstehende Operationen befinden, die Sie abbrechen möchten. Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.
4. Wählen Sie in der Cluster-Liste den Namen des Clusters, nicht das Kästchen links vom Namen des Clusters, der ausstehende Operationen hat, die Sie abbrechen möchten.
5. Um festzustellen, welche Vorgänge noch nicht abgeschlossen sind, wählen Sie die Registerkarte Beschreibung und prüfen Sie, wie viele ausstehende Erstellungen oder Löschungen angezeigt werden. Sie können nicht sowohl ausstehende Erstellungen als auch ausstehende Löschungen haben.
6. Wählen Sie die Registerkarte Nodes aus.
7. Um alle anstehenden Operationen zu stornieren, klicken Sie auf Cancel Pending. Das Dialogfeld Cancel Pending wird angezeigt.
8. Bestätigen Sie mit der Schaltfläche Cancel Pending, dass sie alle anstehenden Operationen stornieren möchten. Wenn die Operationen beibehalten werden sollen, wählen Sie Cancel.

Löschen eines Clusters in ElastiCache

Solange sich ein ElastiCache Cluster im Status „Verfügbar“ befindet, wird er Ihnen in Rechnung gestellt, unabhängig davon, ob Sie ihn aktiv nutzen oder nicht. Löschen Sie den Cluster, damit keine Gebühren mehr anfallen.

Warning

Wenn Sie einen ElastiCache Cluster löschen, werden Ihre manuellen Snapshots beibehalten. Sie können auch einen letzten Snapshot erstellen, bevor der Cluster gelöscht wird. Automatisch erstellte Cache-Snapshots werden nicht aufbewahrt.

Unter Verwendung der AWS Management Console

Mit dem folgenden Verfahren wird ein einzelner Cluster aus Ihrer Bereitstellung gelöscht. Um mehrere Cache-Cluster zu löschen, wiederholen Sie das Verfahren für jeden Cluster, den Sie löschen möchten. Sie brauchen nicht zu warten, bis ein Cluster fertig gelöscht ist, bevor Sie den Vorgang zum Löschen eines anderen Clusters starten.

Löschen eines Clusters

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Engine-Dashboard die Engine aus, auf der der Cluster, den Sie löschen möchten, ausgeführt wird.

Eine Liste mit allen Clustern, in denen diese Engine läuft, wird angezeigt.

3. Wählen Sie aus der Liste den Namen des zu löschenden Clusters aus.

Important

Sie können jeweils nur einen Cluster von der ElastiCache Konsole aus löschen. Werden mehrere Cluster ausgewählt, wird die Löschoption deaktiviert.

4. Klicken Sie bei Actions auf Delete.
5. Wählen Sie auf dem Bestätigungsbildschirm Delete Cluster (Cluster löschen) die Option Delete (Löschen), um den Cluster zu löschen, oder Cancel (Abbrechen), um den Cluster beizubehalten.

Wenn Sie Delete auswählen, ändert sich der Status des Clusters zu deleting.

Sobald dieser Cluster nicht mehr in der Cluster-Liste erscheint, fallen dafür keine Gebühren mehr an.

Verwenden Sie den AWS CLI , um einen ElastiCache Cluster zu löschen

Der folgende Code löscht den ElastiCache Cache-Cluster `my-cluster`.

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

Die `delete-cache-cluster` CLI Aktion löscht nur einen Cache-Cluster. Um mehrere Cache-Cluster zu löschen, rufen Sie `delete-cache-cluster` für jeden Cache-Cluster auf, den Sie löschen möchten. Wenn Sie Ihren Cluster in einer VPC starten, die auf dem `-Service` basiert, können Sie außerhalb von darauf zugreifen.

Für Linux, macOS oder Unix:

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

Für Windows:

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

Weitere Informationen finden Sie im AWS CLI ElastiCache For-Thema [delete-cache-cluster](#).

Mit dem ElastiCache API

Der folgende Code löscht den 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>
```

Der `DeleteCacheCluster` API Vorgang löscht nur einen Cache-Cluster. Um mehrere Cache-Cluster zu löschen, rufen Sie `DeleteCacheCluster` für jeden Cache-Cluster auf, den Sie löschen möchten. Wenn Sie Ihren Cluster in einer VPC starten, die auf dem `-Service` basiert, können Sie außerhalb von darauf zugreifen.

Weitere Informationen finden Sie im ElastiCache API Referenzthema [DeleteCacheCluster](#).

Zugreifen auf Ihren ElastiCache Cluster oder Ihre Replikationsgruppe

Ihre ElastiCache Amazon-Instances sind für den Zugriff über eine EC2 Amazon-Instance konzipiert.

Wenn Sie Ihre ElastiCache Instance in einer Amazon Virtual Private Cloud (AmazonVPC) gestartet haben, können Sie von einer ElastiCache EC2 Amazon-Instance in demselben Amazon aus auf Ihre Instance zugreifenVPC. Oder Sie können mithilfe von VPC Peering von einem Amazon EC2 in einem anderen Amazon VPC auf Ihre ElastiCache Instance zugreifen.

Wenn Sie Ihre ElastiCache Instance in EC2 Classic gestartet haben, erlauben Sie der EC2 Instance den Zugriff auf Ihren Cluster, indem Sie der mit der Instance verknüpften EC2 Amazon-Sicherheitsgruppe Zugriff auf Ihre Cache-Sicherheitsgruppe gewähren. Standardmäßig ist der Zugriff auf einen Cluster auf das Konto beschränkt, mit dem der Cluster gestartet wurde.

Themen

- [Zugriff auf Ihren Cluster oder die Replikationsgruppe erteilen](#)

Zugriff auf Ihren Cluster oder die Replikationsgruppe erteilen

Sie haben Ihren Cluster gestartet in EC2 - VPC

Wenn Sie Ihren Cluster in einer Amazon Virtual Private Cloud (AmazonVPC) gestartet haben, können Sie nur von einer EC2 Amazon-Instance aus, die in demselben Amazon läuft, eine Verbindung zu Ihrem ElastiCache Cluster herstellenVPC. In diesem Fall müssen Sie Netzwerkzugang zum Cluster gewähren.

Note

Wenn Sie Local Zones verwenden, vergewissern Sie sich, dass Sie sie aktiviert haben. Weitere Informationen finden Sie unter [Local Zones aktivieren](#). Auf diese Weise VPC VPC wird Ihr Subnetz auf diese lokale Zone ausgedehnt und Sie behandeln das Subnetz wie jedes andere Subnetz in jeder anderen Availability Zone, und die relevanten Gateways, Routing-Tabellen und andere Überlegungen zu Sicherheitsgruppen werden automatisch angepasst.

So gewähren Sie Netzwerkzugang von einer VPC Amazon-Sicherheitsgruppe zu einem Cluster

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.

2. Wählen Sie in der Navigationsleiste unter Network & Security die Option Security Groups aus.
3. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe für Ihr Amazon aus VPC. Sofern Sie keine Sicherheitsgruppe zur ElastiCache Verwendung erstellt haben, wird diese Sicherheitsgruppe als Standard bezeichnet.
4. Wählen Sie die Registerkarte Inbound und verfahren Sie dann wie folgt:
 - a. Wählen Sie Edit aus.
 - b. Wählen Sie Regel hinzufügen aus.
 - c. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP Regel aus.
 - d. Geben Sie in das Feld Port Range die Portnummer Ihres Clusterknotens ein. Diese Nummer muss mit der Nummer übereinstimmen, die Sie beim Starten des Clusters angegeben haben. Der Standardport für Memcached ist **11211** Der Standardport für Valkey und Redis ist. OSS **6379**
 - e. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrem Amazon starten, eine Verbindung zu Ihren VPC Knoten herstellen kann. ElastiCache

 **Important**

Wenn Sie den ElastiCache Cluster für 0.0.0.0/0 öffnen, wird der Cluster nicht dem Internet zugänglich gemacht, da er keine öffentliche IP-Adresse hat und daher nicht von außerhalb des darauf zugegriffen werden kann. VPC Die Standardsicherheitsgruppe kann jedoch auf andere EC2 Amazon-Instances im Kundenkonto angewendet werden, und diese Instances können eine öffentliche IP-Adresse haben. Wenn diese Instances zufälligerweise eine Aktion auf dem Standardport ausführen, dann könnte dieser Service unbeabsichtigt zugänglich gemacht werden. Daher empfehlen wir, eine VPC Sicherheitsgruppe zu erstellen, die ausschließlich von verwendet wird ElastiCache. Weitere Informationen finden Sie unter [Benutzerdefinierte Sicherheitsgruppen](#).

- f. Wählen Sie Save (Speichern) aus.

Wenn Sie eine EC2 Amazon-Instance in Ihrem Amazon starten VPC, kann diese Instance eine Verbindung zu Ihrem ElastiCache Cluster herstellen.

Zugriff auf ElastiCache Ressourcen von außen AWS

Amazon ElastiCache ist ein AWS Service, der einen cloudbasierten In-Memory-Key-Value-Speicher bereitstellt. Der Service ist so konzipiert, dass er ausschließlich von innen abgerufen werden kann. Wenn der ElastiCache Cluster jedoch in einer VPC gehostet wird, können Sie eine Network Address Translation (NAT) -Instanz verwenden, um den Zugriff von außen zu ermöglichen.

Voraussetzungen

Die folgenden Anforderungen müssen erfüllt sein, damit Sie von außerhalb auf Ihre ElastiCache Ressourcen zugreifen können AWS:

- Der Cluster muss sich in einer Network Address Translation (NAT) -Instanz befinden VPC und der Zugriff darauf ist über eine Network Address Translation (NAT) -Instanz möglich. Für diese Anforderung gibt es keine Ausnahmen.
- Die NAT Instanz muss in derselben Weise VPC wie der Cluster gestartet werden.
- Die NAT Instance muss in einem vom Cluster getrennten öffentlichen Subnetz gestartet werden.
- Der NAT Instance muss eine Elastic IP-Adresse (EIP) zugeordnet sein. Die Portweiterleitungsfunktion von iptables wird verwendet, um einen Port auf der NAT Instance an den Cache-Node-Port innerhalb von weiterzuleiten. VPC

Überlegungen

Die folgenden Überlegungen sollten berücksichtigt werden, wenn Sie von außen ElastiCache auf Ihre ElastiCache Ressourcen zugreifen.

- Clients stellen eine Verbindung zum EIP und zum Cache-Port der NAT Instance her. Die Portweiterleitung auf der NAT Instance leitet den Datenverkehr an den entsprechenden Cache-Cluster-Knoten weiter.
- Wenn ein Cluster-Knoten hinzugefügt oder ersetzt wird, müssen die iptables-Regeln entsprechend aktualisiert werden.

Einschränkungen

Dieser Ansatz sollte nur zu Test- und Entwicklungszwecken verwendet werden. Er wird aufgrund der folgenden Einschränkungen nicht für die Produktion empfohlen:

- Die NAT Instanz fungiert als Proxy zwischen Clients und mehreren Clustern. Ein zusätzlicher Proxy beeinträchtigt die Leistung des Cache-Clusters. Die Auswirkung nimmt mit der Anzahl der Cache-Cluster zu, auf die Sie über die NAT Instance zugreifen.
- Der Datenverkehr von den Clients zur NAT Instance ist unverschlüsselt. Daher sollten Sie vermeiden, sensible Daten über die NAT Instance zu senden.
- Die NAT Instanz erhöht den Mehraufwand für die Wartung einer anderen Instanz.
- Die NAT Instanz dient als zentrale Fehlerquelle. Informationen zur Einrichtung von Hochverfügbarkeit NAT auf VPC finden Sie unter [Hochverfügbarkeit für VPC NAT Amazon-Instances: Ein Beispiel](#).

Wie greife ich von außen auf ElastiCache Ressourcen zu AWS

Das folgende Verfahren zeigt, wie Sie mithilfe einer NAT Instanz eine Verbindung zu Ihren ElastiCache Ressourcen herstellen.

Diese Schritte setzen Folgendes voraus:

- `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`

Als Nächstes benötigen Sie NAT in die entgegengesetzte Richtung:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

Sie müssen auch die IP-Weiterleitung aktivieren, die standardmäßig deaktiviert ist:

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf sudo sysctl --system
```

- Ihr Zugriff auf einen Memcached-Cluster erfolgt mit:
 - IP-Adresse – 10.0.1.230
 - Memcached-Standardport – 11211
 - Sicherheitsgruppe – *10\0\0\55*
- Sie greifen auf einen Valkey- oder OSS Redis-Cluster zu mit:

- IP-Adresse – 10.0.1.230
- Standardport — 6379
- Sicherheitsgruppe – sg-bd56b7da
- AWS Instanz-IP-Adresse — sg-bd56b7da
- Die IP-Adresse Ihres vertrauenswürdigen Clients lautet 198.51.100.27.
- Ihre NAT Instance hat die Elastic IP-Adresse 203.0.113.73.
- Ihre NAT Instance hat die Sicherheitsgruppe sg-ce56b7a9.

So stellen Sie mithilfe einer Instance eine Verbindung zu Ihren Ressourcen her ElastiCache NAT

1. Erstellen Sie eine NAT Instanz im selben VPC wie Ihr Cache-Cluster, jedoch in einem öffentlichen Subnetz.

Standardmäßig startet der VPC Assistent den Knotentyp `cache.m1.small`. Wählen Sie eine Knotengröße aus, die Ihren Anforderungen entspricht. Sie müssen verwenden EC2 NATAMI, um von außen darauf zugreifen zu können. ElastiCache AWS

Informationen zum Erstellen einer NAT Instanz finden Sie im AWS VPC Benutzerhandbuch unter [NATInstanzen](#).

2. Erstellen Sie Sicherheitsgruppenregeln für den Cache-Cluster und die NAT Instanz.

Die NAT Instanz-Sicherheitsgruppe und die Clusterinstanz sollten die folgenden Regeln haben:

- Zwei Regeln für eingehenden Datenverkehr
 - Bei Memcached besteht die erste Regel darin, TCP Verbindungen von vertrauenswürdigen Clients zu jedem Cache-Port zuzulassen, der von der NAT Instance weitergeleitet wird (11211 — 11213).
 - Bei Valkey und Redis besteht die erste Regel darin, TCP Verbindungen von vertrauenswürdigen Clients zu jedem Cache-Port zuzulassen, der von der NAT Instanz weitergeleitet wird (6379 — 6381).
 - Eine zweite Regel ermöglicht den SSH Zugriff auf vertrauenswürdige Clients.

NATInstanz-Sicherheitsgruppe — Regeln für eingehenden Datenverkehr mit Memcached

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
TCPBenutz erdefinierte Regel	TCP	11211 –11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

NATInstanz-Sicherheitsgruppe — Regeln für eingehenden Datenverkehr mit Valkey oder Redis OSS

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
TCPBenutz erdefinierte Regel	TCP	6379-6380	198.51.100.27/32
SSH	TCP	22	203.0.113.73/32

- Bei Memcached handelt es sich um eine ausgehende Regel, die TCP Verbindungen zum Cache-Port (11211) zulässt.

NATInstanz-Sicherheitsgruppe — ausgehende Regel

Typ	Protocol (Protokoll)	Port-Bereich	Bestimmungsort
Benutzerdefinierte Regel TCP	TCP	11211	sg-ce56b7 a9 (Sicherheitsgruppe der Cluster-Instance)

- Mit Valkey oder RedisOSS, einer ausgehenden Regel, die TCP Verbindungen zum Cache-Port (6379) zulässt.

NATInstanz-Sicherheitsgruppe — ausgehende Regel

Typ	Protocol (Protokoll)	Port-Bereich	Bestimmungsort
Benutzerdefinierte Regel TCP	TCP	6379	sg-ce56b7a9 (Sicherheitsgruppe der Cluster-Instance)

- Mit Memcached, einer eingehenden Regel für die Sicherheitsgruppe des Clusters, die TCP Verbindungen von der NAT Instance zum Cache-Port (11211) zulässt.

Sicherheitsgruppe der Cluster-Instance – Regel für eingehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Benutzerdefinierte Regel TCP	TCP	11211	sg-bd56b7da (Sicherheitsgruppe) NAT

- Mit Valkey oder RedisOSS, einer eingehenden Regel für die Sicherheitsgruppe des Clusters, die TCP Verbindungen von der Instance zum Cache-Port (6379) zulässt. NAT

Sicherheitsgruppe der Cluster-Instance – Regel für eingehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Benutzerdefinierte Regel TCP	TCP	6379	sg-bd56b7da (Cluster-Sicherheitsgruppe)

3. Validieren Sie die Regeln.

- Vergewissern Sie sich, dass der vertrauenswürdige Client Zugriff SSH auf die NAT Instanz hat.
- Vergewissern Sie sich, dass der vertrauenswürdige Client von der NAT Instanz aus eine Verbindung zum Cluster herstellen kann.

4. Memcached

Fügen Sie der Instanz eine iptables-Regel hinzu. NAT

Für jeden Knoten im Cluster muss der NAT Tabelle eine iptables-Regel hinzugefügt werden, um den Cache-Port von der NAT Instanz an den Clusterknoten weiterzuleiten. Ein Beispiel könnte folgendermaßen aussehen:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
10.0.1.230:11211
```

Die Portnummer muss für jeden Knoten im Cluster eindeutig sein. Wenn Sie z. B. mit einem aus drei Knoten bestehenden Memcached-Cluster unter Verwendung der Ports 11211 – 11213 arbeiten, sehen die Regeln wie folgt aus:

```
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
```

Bestätigen Sie, dass der vertrauenswürdige Client eine Verbindung mit dem Cluster herstellen kann.

Der vertrauenswürdige Client sollte eine Verbindung zu dem mit der NAT Instanz EIP verknüpften und dem Cluster-Port herstellen, der dem entsprechenden Clusterknoten entspricht. Die Verbindungszeichenfolge für PHP könnte beispielsweise wie folgt aussehen:

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

Zum Überprüfen der Verbindung kann auch ein Telnet-Client verwendet werden. Beispielsweise:

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

Valkey oder Redis OSS

Fügen Sie der Instanz eine iptables-Regel hinzu. NAT

Für jeden Knoten im Cluster muss der NAT Tabelle eine iptables-Regel hinzugefügt werden, um den Cache-Port von der NAT Instanz an den Clusterknoten weiterzuleiten. Ein Beispiel könnte folgendermaßen aussehen:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6379 -j DNAT --to
10.0.1.230:6379
```

Die Portnummer muss für jeden Knoten im Cluster eindeutig sein. Wenn Sie beispielsweise mit einem OSS Redis-Cluster mit drei Knoten arbeiten und die Ports 6379 — 6381 verwenden, würden die Regeln wie folgt aussehen:

```
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
```

Bestätigen Sie, dass der vertrauenswürdige Client eine Verbindung mit dem Cluster herstellen kann.

Der vertrauenswürdige Client sollte eine Verbindung zu dem der EIP NAT Instanz und dem Cluster-Port herstellen, der dem entsprechenden Clusterknoten entspricht. Die Verbindungszeichenfolge für PHP könnte beispielsweise wie folgt aussehen:

```
redis->connect( '203.0.113.73', 6379 );
redis->connect( '203.0.113.73', 6380 );
redis->connect( '203.0.113.73', 6381 );
```

Zum Überprüfen der Verbindung kann auch ein Telnet-Client verwendet werden. Beispielsweise:

```
telnet 203.0.113.73 6379
telnet 203.0.113.73 6380
telnet 203.0.113.73 6381
```

5. Speichern Sie die iptables-Konfiguration.

Speichern Sie die Regeln, nachdem Sie sie getestet und überprüft haben. Wenn Sie eine Redhat-Linux-Verteilung (wie Amazon Linux) verwenden, führen Sie den folgenden Befehl aus:

```
service iptables save
```

Verwandte Themen

Die folgenden Themen können von zusätzlichem Interesse sein.

- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC](#)
- [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird](#)
- [NATInstanzen](#)
- [Konfiguration von ElastiCache Clients](#)
- [Hochverfügbarkeit für VPC NAT Amazon-Instances: Ein Beispiel](#)

Verbindungsendpunkte finden in ElastiCache

Ihre Anwendung stellt über Endpunkte eine Verbindung zu Ihrem ElastiCache Cluster her. Ein Endpunkt ist die eindeutige Adresse eines Knotens oder Clusters.

Welche Endpunkte sollen mit Valkey oder Redis verwendet werden OSS

- eigenständiger Knoten, verwendet den Endpunkt des Knotens sowohl für Lese- als auch für Schreiboperationen.
- Valkery-, Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) verwenden den primären Endpunkt für alle Schreibvorgänge. Verwenden Sie Leser-Endpunkt, um am Endpunkt ankommende Verbindungen auf alle Lesereplikate zu verteilen. Verwenden Sie die einzelnen Node-Endpoints für Lesevorgänge (im FeldAPI/werden CLI diese als Lese-Endpunkte bezeichnet).
- Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) verwenden den Konfigurationsendpunkt des Clusters für alle Operationen, die Befehle mit aktiviertem Clustermodus unterstützen. Sie müssen einen Client verwenden, der Valkey Cluster oder Redis OSS Cluster (Redis 3.2) unterstützt. OSS Sie können immer noch von einzelnen Knotenendpunkten lesen (In der DateiAPI/werden CLI diese als Read-Endpunkte bezeichnet).

Die folgenden Abschnitte führen Sie durch die Erkennung der Endpunkte, die Sie für die von Ihnen ausgeführte Engine benötigen.

Welche Endpunkte sollen mit Memcached verwendet werden

Für ElastiCache serverlosen Cache mit Memcached rufen Sie einfach den Cluster-Endpunkt DNS und den Port von der Konsole ab.

Verwenden Sie von der aus den `describe-serverless-caches` Befehl AWS CLI, um die Endpunktinformationen abzurufen.

Linux

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

Windows

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

Die Ausgabe der obigen Operation sollte ungefähr so aussehen (JSONFormat):

```
{
  "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"
}
]
```

Instance-basierter Memcached-Cluster – Wenn Sie Automatic Discovery verwenden, können Sie Ihren Memcached-Client über den Konfigurationsendpunkt des Clusters konfigurieren. Dies bedeutet, dass Sie einen Client verwenden müssen, der Automatic Discovery unterstützt.

Wenn Sie Automatic Discovery nicht verwenden, müssen Sie den Client so konfigurieren, dass er die einzelnen Knotenendpunkte für Lese- und Schreibvorgänge verwendet. Auch müssen Sie sie beim Hinzufügen und Entfernen von Knoten verfolgen.

Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS (Cluster-Modus deaktiviert) (Konsole)

Wenn ein Valkey-, Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) nur einen Knoten hat, wird der Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge verwendet. Wenn ein Valkey-, Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) mehrere Knoten hat, gibt es drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte.

Der primäre Endpunkt ist ein DNS Name, der immer zum primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Leser-Endpunkt teilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem ElastiCache (Redis OSS -) Cluster auf. Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache (Redis-OSS) Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Leser-Endpunkte sicherzustellen.

Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS Datensatz, der nach dem Round-Robin-Verfahren in die IP-Adresse eines der Replikatknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

So finden Sie die Endpunkte eines Valkey-, Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Konsole unter ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.

Der Cluster-Bildschirm wird mit einer Liste von Valkery- oder Valkey- oder Redis-Clustern OSS (Cluster-Modus deaktiviert) und Valkey- oder Redis-Clustern (Cluster-Modus aktiviert) angezeigt. OSS

- Um den primären und/oder den Reader-Endpunkt des Clusters zu ermitteln, wählen Sie den Namen des Clusters aus (nicht die Schaltfläche links davon).

▼ 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 -encrypted.llru6f.ng.0001.use1.cache.ama zonaws.com:6379	Reader endpoint -encrypted-ro.llru6f.ng.0001.use1.cache.a mazonaws.com:6379	ARN	

Primäre Endpunkte und Reader-Endpunkte für einen Valkery-, Valkey- oder Redis-Cluster (Clustermodus deaktiviert) OSS

Wenn sich im Cluster nur ein Knoten befindet, ist kein primärer Endpunkt vorhanden. Sie können mit dem nächsten Schritt fortfahren.

- Wenn der Valkery-, Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) über Replikatknoten verfügt, können Sie die Replikatknotenendpunkte des Clusters finden, indem Sie den Namen des Clusters und dann die Registerkarte Knoten auswählen.

Auf dem angezeigten Knotenbildschirm wird jeder Knoten im Cluster, –primärer Knoten und Replikate, – mit seinem Endpunkt aufgelistet.

<input type="checkbox"/>	Node Name	Status	Current Role	Port	Endpoint
<input type="checkbox"/>	test-no-001	available	primary	6379	-encrypted.llru6f.ng.0001.use1.cache.ama zonaws.com:6379
<input type="checkbox"/>	test-no-002	available	replica	6379	-encrypted-ro.llru6f.ng.0001.use1.cache.a mazonaws.com:6379
<input type="checkbox"/>	test-no-003	available	replica	6379	-encrypted-ro.llru6f.ng.0001.use1.cache.a mazonaws.com:6379

Knotenendpunkte für einen Valkery-, Valkey- oder Redis-Cluster (Clustermodus deaktiviert) OSS

5. So kopieren Sie einen Endpunkt in Ihre Zwischenablage:
 - a. Suchen Sie einen Endpunkt nach dem anderen nach dem zu kopierenden Endpunkt.
 - b. Wählen Sie das Kopiersymbol direkt vor dem Endpunkt aus.

Der Endpunkt wird nun in Ihre Zwischenablage kopiert. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Verbindung zu Memcached-Knoten herstellen](#).

Ein primärer Endpunkt von Valkery, Valkey oder Redis OSS (Clustermodus deaktiviert) sieht etwa wie folgt aus. Abhängig davon, ob die Verschlüsselung während der Übertragung aktiviert ist oder nicht, unterscheidet sich dies.

Verschlüsselung während der Übertragung nicht aktiviert

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

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

Verschlüsselung während der Übertragung aktiviert

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

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

Finden von Endpunkten für einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) (Konsole)

Ein Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) hat einen einzigen Konfigurationsendpunkt. Über die Verbindung zum Konfigurationsendpunkt findet die Anwendung für jede Shard im Cluster die primären Endpunkte sowie jene für Schreibvorgänge.

Um den Endpunkt eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) zu finden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.

Der Cluster-Bildschirm mit einer Liste von Clustern wird angezeigt. Wählen Sie den Cluster aus, zu dem Sie eine Verbindung herstellen möchten.

- Um den Konfigurationsendpunkt des Clusters zu suchen, wählen Sie den Namen des Clusters (nicht das Optionsfeld) aus.
- Der Configuration endpoint (Konfigurationsendpunkt) wird unter Cluster details (Cluster-Details) angezeigt. Wählen Sie zum Kopieren das Symbol Copy (Kopieren) links vom Endpunkt aus.

Die Endpunkte eines Clusters finden (Konsole) (Memcached)

Alle Memcached-Endpunkte sind Lese-/Schreib-Endpunkte. Zum Herstellen einer Verbindung mit Knoten in einem Memcached-Cluster kann Ihre Anwendung entweder die Endpunkte für jeden Knoten oder den Konfigurationsendpunkt des Clusters zusammen mit Automatic Discovery verwenden. Um Automatic Discovery verwenden zu können, benötigen Sie einen Client, der Automatic Discovery unterstützt.

Wenn Automatic Discovery verwendet wird, stellt Ihre Client-Anwendung über den Konfigurationsendpunkt eine Verbindung mit Ihrem Memcached-Cluster her. Während Sie Ihren Cluster durch Hinzufügen oder Entfernen von Knoten skalieren, „kennt“ Ihre Anwendung automatisch alle Knoten im Cluster und kann zu beliebigen von ihnen eine Verbindung herstellen. Ohne Automatic Discovery wäre das Ihre Aufgabe oder Sie müssten die Endpunkte Ihrer Anwendung jedes Mal manuell aktualisieren, wenn Sie einen Knoten hinzufügen oder löschen würden.

Wenn Sie einen Endpunkt kopieren möchten, wählen Sie das Kopiersymbol direkt vor der Endpunktadresse aus. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Verbindung zu Memcached-Knoten herstellen](#).

Die Konfiguration und die Knotenendpunkte sehen sehr ähnlich aus. Es folgen die mit Fettdruck hervorgehobenen Unterschiede.

```
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

Wenn Sie einen Konfigurationsendpunkt CNAME für Ihren Memcached erstellen möchten, damit Ihr Automatic Discovery Client ihn CNAME als Konfigurationsendpunkt erkennt, müssen Sie in den `.cfg` CNAME

Suchen von Endpunkten (AWS CLI)

Für Memcached können Sie AWS CLI for Amazon verwenden, ElastiCache um die Endpunkte für Knoten und Cluster zu ermitteln.

Für Redis können Sie AWS CLI for Amazon verwenden, ElastiCache um die Endpunkte für Knoten, Cluster und auch Replikationsgruppen zu ermitteln.

Themen

- [Suchen von Endpunkten für Knoten und Cluster \(AWS CLI\)](#)
- [Finden der Endpunkte für Valkey- oder OSS Redis-Replikationsgruppen \(\)AWS CLI](#)

Suchen von Endpunkten für Knoten und Cluster (AWS CLI)

Sie können den verwenden AWS CLI , um die Endpunkte für einen Cluster und seine Knoten mit dem Befehl zu ermitteln. `describe-cache-clusters` Für Valkey- oder OSS Redis-Cluster gibt der Befehl den Cluster-Endpunkt zurück. Für Memcached-Cluster gibt der Befehl den Konfigurationsendpunkt zurück. Wenn Sie den optionalen Parameter `--show-cache-node-info` einschließen, gibt der Befehl auch die Endpunkte der einzelnen Knoten im Cluster zurück.

Example

Der folgende Befehl ruft den Konfigurationsendpunkt (`ConfigurationEndpoint`) und einzelne Knotenendpunkte (`Endpoint`) für den Memcached-Cluster `mycluster` ab.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

Die Ausgabe des obigen Vorgangs sollte ungefähr so aussehen (JSONFormat).

```
{
```

```
"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

Wenn Sie einen Konfigurationsendpunkt CNAME für Ihren Memcached erstellen möchten, damit Ihr Auto Discovery-Client ihn CNAME als Konfigurationsendpunkt erkennt, müssen Sie ihn `.cfg.` in den aufnehmen. CNAME Zum Beispiel `mycluster.cfg.local` in Ihrer `php.ini`-Datei für den Parameter `session.save_path`.

Example

Für Valkey und Redis OSS ruft der folgende Befehl die Clusterinformationen für den Einzelknotencluster `mycluster` ab.

Important

Der Parameter `--cache-cluster-id` kann mit der Cluster-ID Valkey oder Redis OSS (Clustermodus deaktiviert) mit einem Knoten oder mit bestimmten Knoten-IDs in Replikationsgruppen verwendet werden. Der Wert `--cache-cluster-id` einer Replikationsgruppe ist ein vierstelliger Wert, wie z. B. `0001` Wenn `--cache-cluster-id` es

sich um die ID eines Clusters (Knotens) in einer Replikationsgruppe handelt, `replication-group-id` ist der in der Ausgabe enthalten.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id redis-cluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id redis-cluster ^  
  --show-cache-node-info
```

Die Ausgabe der obigen Operation sollte ungefähr so aussehen (JSONFormat).

```
{  
  "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"
  }
]
}

```

Weitere Informationen finden Sie im Thema [describe-cache-clusters](#).

Finden der Endpunkte für Valkey- oder OSS Redis-Replikationsgruppen ()AWS CLI

Sie können den verwenden AWS CLI , um die Endpunkte für eine Replikationsgruppe und ihre Cluster mit dem Befehl zu ermitteln. `describe-replication-groups` Der Befehl gibt den primären Endpunkt der Replikationsgruppe und eine Liste aller Cluster (Knoten) in der Replikationsgruppe mit ihren Endpunkten sowie den Reader-Endpunkt zurück.

Die folgende Operation ruft den primären Endpunkt und den Reader-Endpunkt der Replikationsgruppe `myreplgroup` ab. Verwenden Sie den primären Endpunkt für alle Schreibvorgänge.

```

aws elasticache describe-replication-groups \
  --replication-group-id myreplgroup

```

Für Windows:

```
aws elasticache describe-replication-groups ^  
  --replication-group-id myreplgroup
```

Die Ausgabe dieses Vorgangs sollte ungefähr so aussehen (JSONFormat).

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

Weitere Informationen finden Sie [describe-replication-groups](#) in der AWS CLI Befehlsreferenz.

Endpunkte finden () ElastiCache API

Für Memcached können Sie Amazon verwenden, ElastiCache API um die Endpunkte für Knoten und Cluster zu ermitteln.

Für Redis können Sie Amazon verwenden, ElastiCache API um die Endpunkte für Knoten, Cluster und auch Replikationsgruppen zu ermitteln.

Themen

- [Endpunkte für Knoten und Cluster finden \(\) ElastiCache API](#)
- [Suchen nach Endpunkten für Valkey- oder OSS Redis-Replikationsgruppen \(\) ElastiCache API](#)

Endpunkte für Knoten und Cluster finden () ElastiCache API

Mit der ElastiCache API Aktion können Sie die Endpunkte für einen Cluster und seine Knoten ermitteln. `DescribeCacheClusters` Für Valkey- oder OSS Redis-Cluster gibt der Befehl den Cluster-Endpoint zurück. Für Memcached-Cluster gibt der Befehl den Konfigurationsendpoint zurück. Wenn Sie den optionalen Parameter `ShowCacheNodeInfo` einschließen, gibt die Aktion auch die Endpunkte der einzelnen Knoten im Cluster zurück.

Example

Für Memcached ruft der folgende Befehl den Konfigurationsendpoint (*ConfigurationEndpoint*) und die einzelnen Knotenendpunkte (*Endpoint*) für den Memcached-Cluster `mycluster` ab.

```
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

Wenn Sie einen Konfigurationsendpoint CNAME für Ihren Memcached erstellen möchten, damit Ihr Auto Discovery-Client ihn CNAME als Konfigurationsendpoint erkennt, müssen

Sie ihn `.cfg.` in den aufzunehmen. CNAME Zum Beispiel `mycluster.cfg.local` in Ihrer `php.ini`-Datei für den Parameter `session.save_path`.

Suchen nach Endpunkten für Valkey- oder OSS Redis-Replikationsgruppen () ElastiCache API

Mit der Aktion können Sie ElastiCache API die Endpunkte für eine Replikationsgruppe und ihre Cluster ermitteln. `DescribeReplicationGroups` Die Aktion gibt den primären Endpunkt der Replikationsgruppe und eine Liste aller Cluster in der Replikationsgruppe mit ihren Endpunkten sowie den Reader-Endpunkt zurück.

Mit dem folgenden Vorgang werden der primäre Endpunkt (`PrimaryEndpoint`), der Leser-Endpunkt (`ReaderEndpoint`) und die Endpunkte einzelner Knoten (`ReadEndpoint`) für die Replikationsgruppe abgerufen. `myreplgroup` Verwenden Sie den primären Endpunkt für alle Schreibvorgänge.

```
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>
```

Weitere Informationen finden Sie unter. [DescribeReplicationGroups](#)

Arbeiten mit Shards in ElastiCache

Ein Shard (API/CLI: Node Group) ist eine Sammlung von eins bis sechs ElastiCache mit Valkey- oder Redis-Knoten. OSS Ein Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) wird nie mehr als einen Shard haben. Mit Shards können Sie große Datenbanken in kleinere, schnellere und einfacher zu verwaltende Teile aufteilen, die als Daten-Shards bezeichnet werden. Dadurch kann die Datenbankeffizienz erhöht werden, indem die Operationen auf mehrere separate Abschnitte verteilt werden. Die Verwendung von Shards kann viele Vorteile bieten, darunter eine verbesserte Leistung, Skalierbarkeit und Kosteneffizienz.

Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die

Anzahl erlaubter Replikate entspricht. Die Daten des Clusters werden über die Shards des Clusters hinweg partitioniert. Falls sich mehr als ein Knoten in einem Shard befindet, implementiert der Shard die Replikation auf dem primären Knoten für Lese- und Schreibvorgänge. Die anderen sind schreibgeschützte Replikatknoten.

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Engine-Version Valkey 7.2 oder Redis 5.0.6 oder höher verwendet OSS wird. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

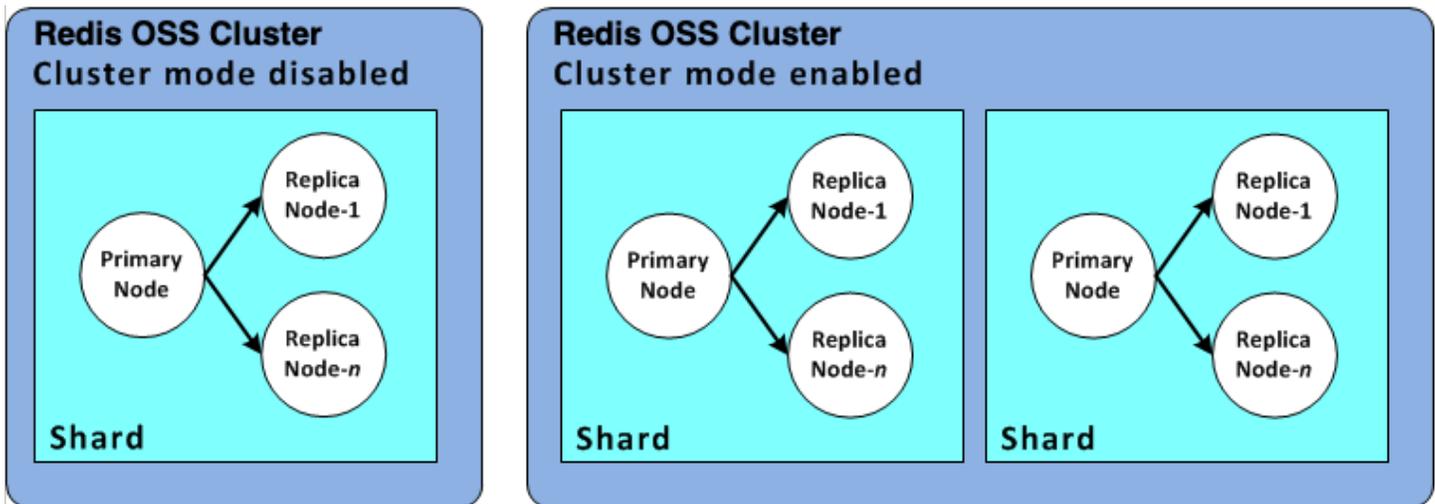
Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, siehe [AWS -Service-Limits](#) und wählen Sie den Limittyp Knoten pro Cluster pro Instance-Typ.

Wenn Sie mit der ElastiCache Konsole einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) erstellen, geben Sie die Anzahl der Shards im Cluster und die Anzahl der Knoten in den Shards an. Weitere Informationen finden Sie unter [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#). Wenn Sie das ElastiCache API oder verwenden AWS CLI, um einen Cluster zu erstellen (im API/als Replikationsgruppe bezeichnet CLI), können Sie die Anzahl der Knoten in einem Shard (API/CLI: Knotengruppe) unabhängig voneinander konfigurieren. Weitere Informationen finden Sie hier:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Für jeden Knoten in einem Shard gelten dieselben Rechner-, Arbeitsspeicher- und Festspeicherspezifikationen. ElastiCache API damit können Sie Shard-weite Attribute steuern, z. B. die Anzahl der Knoten, Sicherheitseinstellungen und Systemwartungsfenster.



Valkey- oder Redis-Shard-Konfigurationen OSS

Weitere Informationen erhalten Sie unter [Offline-Resharding für Valkey oder OSS Redis \(Clustermodus aktiviert\)](#) und [Online-Resharding für Valkey oder OSS Redis \(Clustermodus aktiviert\)](#).

Auffinden der ID eines Shards

Sie können die ID eines Shards mit dem AWS Management Console, dem oder dem finden. AWS CLI ElastiCache API

Mit dem AWS Management Console

Themen

- [Für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#)
- [Für Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)

Für Valkey oder Redis OSS (Cluster-Modus deaktiviert)

Für Valkey oder Redis OSS (Cluster-Modus deaktiviert) gibt es immer Replikationsgruppen-Shard. IDs 0001

Für Valkey oder Redis OSS (Clustermodus aktiviert)

Das folgende Verfahren verwendet die AWS Management Console , um die Shard-ID einer Replikationsgruppe von Valkey oder Redis OSS (Clustermodus aktiviert) zu ermitteln.

So finden Sie die Shard-ID in einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS und dann den Namen der Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) aus, für die Sie den Shard suchen möchten. IDs
3. In der Spalte Shard-Name ist die Shard-ID die letzten vier Ziffern des Shard-Namens.

Verwenden Sie AWS CLI

Um Shard-IDs (Knotengruppen) für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) oder Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus aktiviert) zu finden, verwenden Sie den AWS CLI Vorgang `describe-replication-groups` mit dem folgenden optionalen Parameter.

- **--replication-group-id**—Ein optionaler Parameter, der bei Verwendung die Ausgabe auf die Details der angegebenen Replikationsgruppe beschränkt. Wenn dieser Parameter weggelassen wird, werden die Details von bis zu 100 Replikationsgruppen zurückgegeben.

Example

Dieser Befehl gibt die Details für `sample-repl-group` zurück.

Für Linux, macOS oder Unix:

```
aws elasticache describe-replication-groups \  
  --replication-group-id sample-repl-group
```

Für Windows:

```
aws elasticache describe-replication-groups ^  
  --replication-group-id sample-repl-group
```

Die Ausgabe dieses Befehls sieht in etwa wie folgt aus: Die Shard-IDs (Knotengruppen) lauten *highlighted* hier, um sie leichter zu finden.

```
{
```

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

Mit dem ElastiCache API

Um Shard-IDs (Knotengruppen) für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) oder Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus aktiviert) zu finden, verwenden Sie den AWS CLI Vorgang `describe-replication-groups` mit dem folgenden optionalen Parameter.

- **ReplicationGroupId**—Ein optionaler Parameter, der bei Verwendung die Ausgabe auf die Details der angegebenen Replikationsgruppe beschränkt. Wenn dieser Parameter weggelassen wird, werden die Details von bis zu **xxx** Replikationsgruppen zurückgegeben.

Example

Dieser Befehl gibt die Details für `sample-repl-group` zurück.

Für Linux, macOS oder 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>
```

Vergleich der selbst entworfenen Caches von ValkeyOSS, Redis und Memcached

Amazon ElastiCache unterstützt die Cache-Engines Valkey, Redis OSS und Memcached. Jede Engine hat ihre Vorteile. Anhand der Informationen in diesem Thema können Sie die Engine und Version auswählen, die Ihren Anforderungen am besten entspricht.

Important

Nachdem Sie einen Cache, einen selbst entworfenen Cluster oder eine Replikationsgruppe erstellt haben, können Sie ein Upgrade auf eine neuere Engine-Version durchführen, aber Sie können kein Downgrade auf eine ältere Engine-Version durchführen. Wenn Sie eine ältere Engine-Version verwenden möchten, müssen Sie den vorhandenen Cache, den selbst entworfenen Cluster oder die Replikationsgruppe löschen und ihn mit der früheren Engine-Version erneut erstellen.

Auf den ersten Blick ähneln sich die Engines. Jede Engine befindet sich in einem als Hauptspeicher-basierten Key-Value Store. In der Praxis bestehen jedoch erhebliche Unterschiede.

Wählen Sie Memcached aus, wenn Folgendes auf Sie zutrifft:

- Sie benötigen ein möglichst einfaches Modell.
- Sie müssen große Knoten mit mehreren Kernen oder Threads ausführen.
- Sie müssen Knoten skalieren, hinzufügen und entfernen können, wenn die Anforderungen an Ihr System steigen oder sinken.
- Sie müssen Objekte zwischenspeichern.

Wählen Sie Valkey oder Redis OSS with, ElastiCache wenn Folgendes auf Sie zutrifft:

- ElastiCache mit Valkey 7.2 oder Redis OSS Version 7.0 (erweitert)

[Sie möchten Functions, Sharded Pub/Sub oder Verbesserungen verwenden. ACL](#) Weitere Informationen finden Sie unter [Redis OSS Version 7.0](#) (erweitert).

- ElastiCache (RedisOSS) Version 6.2 (erweitert)

Sie möchten die Möglichkeit haben, Daten zwischen Speicher und SSD unter Verwendung des Knotentyps r6gd umzuschichten. Weitere Informationen finden Sie unter [Daten-Tiering](#).

- ElastiCache (RedisOSS) Version 6.0 (erweitert)

Sie möchten Benutzer mit rollenbasierter Zugriffskontrolle authentifizieren.

Weitere Informationen finden Sie unter [Redis OSS Version 6.0 \(erweitert\)](#).

- ElastiCache (RedisOSS) Version 5.0.0 (erweitert)

Sie möchten [OSSRedis-Streams](#) verwenden, eine Protokollstruktur, die es Produzenten ermöglicht, neue Elemente in Echtzeit anzuhängen, und die es Verbrauchern ermöglicht, Nachrichten entweder blockierend oder nicht blockierend zu konsumieren.

Weitere Informationen finden Sie unter [Redis OSS Version 5.0.0 \(erweitert\)](#).

- ElastiCache (RedisOSS) Version 4.0.10 (erweitert)

Unterstützt sowohl die Verschlüsselung als auch das dynamische Hinzufügen oder Entfernen von Shards zu Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert).

Weitere Informationen finden Sie unter [Redis OSS Version 4.0.10 \(erweitert\)](#).

Die folgenden Versionen sind veraltet, haben das Ende ihrer Lebensdauer erreicht oder stehen kurz vor dem Ende.

- ElastiCache (RedisOSS) Version 3.2.10 (erweitert)

Unterstützt die Fähigkeit, Shards dynamisch zu Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) hinzuzufügen oder daraus zu entfernen.

 **Important**

Derzeit unterstützt ElastiCache (RedisOSS) 3.2.10 keine Verschlüsselung.

Weitere Informationen finden Sie hier:

- [Redis OSS Version 3.2.10 \(erweitert\)](#)

- Bewährte Methoden für das Online-Resharding für RedisOSS. Weitere Informationen finden Sie im Folgenden:
 - [Bewährte Methoden: Online-Resharding](#)
 - [Online-Resharding und Shard-Rebalancing für Valkey oder Redis \(Cluster-Modus aktiviert\) OSS](#)
- [Weitere Informationen zur Skalierung von Redis-Clustern finden Sie unter Skalierung. OSS](#)
- ElastiCache (RedisOSS) Version 3.2.6 (erweitert)

Wenn Sie die Funktionalität früherer OSS Redis-Versionen sowie die folgenden Funktionen benötigen, wählen Sie ElastiCache (OSSRedis) 3.2.6:

- Verschlüsselung während der Übertragung. Weitere Informationen finden Sie unter [Amazon ElastiCache \(RedisOSS\) In-Transit Encryption](#).
- Verschlüsselung im Ruhezustand. Weitere Informationen finden Sie unter [Amazon ElastiCache \(RedisOSS\) At-Rest-Verschlüsselung](#).
- ElastiCache (RedisOSS) (Clustermodus aktiviert) Version 3.2.4

Wenn Sie die Funktionalität von Redis OSS 2.8.x sowie die folgenden Funktionen benötigen, wählen Sie Redis OSS 3.2.4 (Clustermodus):

- Sie müssen Ihre Daten auf zwei bis 500 Knotengruppen verteilen (nur im Cluster-Modus).
- Sie benötigen räumliche Indizierung (Cluster-Modus oder Nicht-Cluster-Modus).
- Sie müssen nicht mehrere Datenbanken unterstützen.
- ElastiCache (RedisOSS) (nicht geclusterter Modus) 2.8.x und 3.2.4 (erweitert)

Wenn Folgendes auf Sie zutrifft, wählen Sie Redis OSS 2.8.x oder Redis 3.2.4 (nicht geclusterter Modus): OSS

- Sie benötigen komplexe Datentypen, wie Zeichenfolgen, Hashes, Listen, Sets, Sorted Sets und Bitmaps.
- Sie müssen Datensätze im Speicher sortieren oder ihnen ein Rang zuweisen.
- Sie benötigen Persistenz für Ihren Schlüsselspeicher.
- Sie müssen Ihre Daten für leseintensive Anwendungen vom Primärknoten in eine oder mehrere Lesereplikate replizieren.
- Sie benötigen ein automatisches Failover, wenn Ihr primärer Knoten ausfällt.

- Sie benötigen Veröffentlichungs- und Subskriptionsfunktionen (Pub/Sub), um Clients über Ereignisse auf dem Server zu informieren.
- Sie benötigen Sicherungs- und Wiederherstellungsfunktionen für selbst entworfene Cluster sowie für serverlose Caches.
- Sie müssen mehrere Datenbanken unterstützen.

Zusammenfassung des Vergleichs von Memcached, Valkey oder Redis OSS (Cluster-Modus deaktiviert) und Valkey oder Redis (Cluster-Modus aktiviert) OSS

	Memcached	Valkey oder Redis (Clustermodus deaktiviert) OSS	Valkey oder Redis OSS (Clustermodus aktiviert)
Engine-Versionen +	1.4.5 und höher	4.0.10 und höher	4.0.10 und höher
Datentypen	Einfach	2.8.x – Complex * Komplex	3.2.x und höher – Komplex
Datenpartitionierung	Ja	Nein	Ja
Modifizierbarer Cluster	Ja	Ja	3.2.10 und höher – Beschränkt
Online-Resharding	Nein	Nein	3.2.10 und höher
Verschlüsselung	während der Übertragung 1.6.12 und höher	4.0.10 und höher	4.0.10 und höher
Daten-Tiering	Nein	6.2 und später	6.2 und später
Compliance-Zertifizierungen			
Compliance-Zertifi- zierung	Ja – 1.6.12 und höher	4.0.10 und höher	4.0.10 und höher
Feed RAMP	Ja – 1.6.12 und höher	4.0.10 und höher	4.0.10 und höher
HIPAA	Ja	4.0.10 und höher	4.0.10 und höher
PCI DSS			
Multi-Thread	Ja	Nein	Nein
Upgrade des Knotentyps	Nein	Ja	Ja

	Memcached	Valkey oder Redis (Clustermodus deaktiviert) OSS	Valkey oder Redis OSS (Clustermodus aktiviert)
Engine-Upgrade	Ja	Ja	Ja
Hohe Verfügbarkeit (Replikation)	Nein	Ja	Ja
Automatisches Failover	Nein	Optional	Erforderlich
Pub-/Sub-Funktionen	Nein	Ja	Ja
Sortierte Sätze	Nein	Ja	Ja
Backup und Wiederherstellung	Nur für serverloses Memcached, nicht für selbst entworfene Memcached-Cluster	Ja	Ja
Räumliche Indizierung	Nein	4.0.10 und höher	Ja

Hinweise:

Zeichenfolge, Objekte (wie Datenbanken)

* Zeichenfolge, Sets, Sorted Sets, Listen, Hashes, Bitmaps, HyperLogLog

Zeichenfolge, Sets, Sorted Sets, Listen, Hashes, Bitmaps, HyperLogLog, räumliche Indizes

+ Schließt Versionen aus, die veraltet sind, das Ende ihrer Nutzungsdauer erreicht haben oder bald ablaufen werden.

Nachdem Sie die Engine für Ihren Cluster ausgewählt haben, verwenden Sie am besten die neueste Version dieser Engine. Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

Online-Migration für Valkey oder Redis OSS

Mithilfe der Online-Migration können Sie Ihre Daten von den selbst gehosteten Open-Source-Diensten Valkey oder Redis OSS auf Amazon zu Amazon migrieren. EC2 ElastiCache

Note

Die Online-Migration zu ElastiCache serverlosen Caches oder Clustern, die auf dem Knotentyp r6gd ausgeführt werden, wird nicht unterstützt.

Übersicht

Um Ihre Daten von Valkey oder Redis, die auf Amazon OSS laufen, als Open-Source-Lösung EC2 zu Amazon zu migrieren, ElastiCache ist eine bestehende oder neu erstellte Amazon-Bereitstellung erforderlich. ElastiCache Die Bereitstellung muss eine migrationsbereite Konfiguration haben. Sie sollte außerdem der gewünschten Konfiguration entsprechen, mit Attributen wie Instance-Typ, Anzahl der Shards und Anzahl der Replikate.

Die Online-Migration ist für die Datenmigration von selbst gehosteten Open-Source-Programmen Valkey oder Redis OSS auf Amazon konzipiert und nicht für EC2 die Übertragung ElastiCache von Daten zwischen Clustern. ElastiCache

Important

Wir empfehlen dringend, die folgenden Abschnitte vollständig zu lesen, bevor Sie den Online-Migrationsprozess starten.

Die Migration beginnt, wenn Sie die Operation oder den `StartMigration` API Befehl aufrufen. AWS CLI Bei der Migration deaktivierter Cluster im Valkey- oder OSS Redis-Clustermodus macht der Migrationsprozess den primären Knoten des Valkey- oder Redis-Clusters zu einem Replikat Ihres primären ElastiCache Quell-Valkey- oder OSS Redis-Clusters. OSS Bei der Migration von OSS Clustern mit aktiviertem Valkey- oder Redis-Clustermodus macht der Migrationsprozess den Primärknoten jedes ElastiCache Shards zu einem Replikat des entsprechenden Shards Ihres Quellclusters, der dieselben Slots besitzt.

Wenn die clientseitigen Änderungen fertig sind, rufen Sie den Vorgang auf. `CompleteMigration` API Durch diesen API Vorgang wird Ihre ElastiCache Bereitstellung auf Ihre primäre Valkey- oder

OSS Redis-Bereitstellung mit Primär- und Replikatknoten (falls zutreffend) hochgestuft. Jetzt können Sie Ihre Client-Anwendung so umleiten, dass sie mit dem Schreiben von Daten beginnt. ElastiCache Während der Migration können Sie den Status der Replikation überprüfen, indem Sie den INFO Befehl [valkey-cli](#) auf Ihren Valkey-Knoten und auf den Primärknoten ausführen. ElastiCache

Schritte zur Migration

Die folgenden Themen beschreiben den Prozess zur Migration der Daten:

- [Vorbereiten Ihrer Quelle und Ihres Ziels für die Migration](#)
- [Testen der Datenmigration](#)
- [Migration starten](#)
- [Verifizieren des Fortschritts der Datenmigration](#)
- [Datenmigration abschließen](#)

Vorbereiten Ihrer Quelle und Ihres Ziels für die Migration

Mit diesen Schritten können Sie sich darauf vorbereiten, Ihre Daten von einer selbst gehosteten Valkey- oder Redis-Quelle auf oder von einem OSS Redis-Cluster EC2 zu ElastiCache einem Valkey-Cluster zu migrieren. ElastiCache

Sie müssen sicherstellen, dass alle vier der unten genannten Voraussetzungen erfüllt sind, bevor Sie die Migration von der Konsole aus starten, oder. ElastiCache API AWS CLI

Um Ihre Quell- und Ziel-Valkey- oder OSS Redis-Knoten für die Migration vorzubereiten

1. Identifizieren Sie die ElastiCache Zielbereitstellung und stellen Sie sicher, dass Sie Daten dorthin migrieren können.

Eine bestehende oder neu erstellte ElastiCache Bereitstellung sollte die folgenden Anforderungen für die Migration erfüllen:

- Es verwendet Valkey oder Redis OSS Engine Version 5.0.6 oder höher.
- Die Verschlüsselung bei der Übertragung ist nicht aktiviert.
- Es hat Multi-AZ aktiviert.
- Es steht ausreichend Speicherplatz für die Daten aus Ihrem Valkey- oder OSS Redis-Cluster zur Verfügung. Zum Konfigurieren der richtigen Einstellungen für reservierten Speicher siehe [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

- Wenn der Clustermodus deaktiviert ist, können Sie direkt von den Valkey- oder OSS Redis-Versionen 2.8.21 auf Valkey oder Redis OSS Version 5.0.6 und höher migrieren, wenn Sie die oder Valkey- oder Redis-Versionen 5.0.6 über die OR-Konsole verwenden. CLI OSS CLI
 - Wenn der Clustermodus aktiviert ist, können Sie direkt von jeder Valkey- oder OSS Redis-Version mit aktiviertem Clustermodus auf OSS Redis-Version 5.0.6 und höher migrieren, wenn Sie die CLI oder OSS Redis-Versionen 5.0.6 über die CLI OR-Konsole verwenden.
 - Die Anzahl der Shards in Quell- und Zieldatei stimmt überein.
 - Ist nicht Teil eines globalen Datenspeichers.
 - Datenklassifizierung ist deaktiviert.
2. Stellen Sie sicher, dass die Konfigurationen Ihres Open-Source-Programms Valkey oder Redis und die Bereitstellung kompatibel sind. OSS ElastiCache

In der ElastiCache Zielbereitstellung sollten mindestens alle folgenden Komponenten mit Ihrer Valkey- oder OSS Redis-Konfiguration für die Replikation kompatibel sein:

- Ihr Cluster sollte nicht aktiviert worden AUTH sein.
 - Die Konfiguration `protected-mode` sollte auf eingestellt seinno.
 - Wenn Sie eine `bind` Konfiguration in Ihrer Valkey- oder OSS Redis-Konfiguration haben, sollte diese aktualisiert werden, um Anfragen von ElastiCache Knoten zuzulassen.
 - Die Anzahl der logischen Datenbanken auf dem ElastiCache Knoten und Ihrem Valkey- oder Redis-Cluster sollte identisch sein. OSS Dieser Wert wird `databases` in der Valkey- oder Redis-Konfiguration festgelegt. OSS
 - Valkey- oder OSS Redis-Befehle, die Datenänderungen vornehmen, sollten nicht umbenannt werden, damit die Daten erfolgreich repliziert werden können. Zum Beispielsync,,,,, undpsync. `info config command cluster`
 - Um die Daten aus Ihrem Valkey- oder OSS Redis-Cluster zu replizieren ElastiCache, stellen Sie sicher, dass ausreichend CPU Speicherplatz vorhanden ist, um diese zusätzliche Last zu bewältigen. Diese Last stammt aus der RDB Datei, die von Ihrem Valkey- oder OSS Redis-Cluster erstellt und über das Netzwerk an den Knoten übertragen wurde. ElastiCache
 - Alle Valkey- oder OSS Redis-Instanzen im Quellcluster sollten auf demselben Port laufen.
3. Stellen Sie sicher, dass Ihre Instances eine Verbindung herstellen können, ElastiCache indem Sie wie folgt vorgehen:
- Stellen Sie sicher, dass die IP-Adresse jeder Instance privat ist.

- Weisen Sie die ElastiCache Bereitstellung derselben virtuellen privaten Cloud (VPC) zu oder erstellen Sie sie in derselben virtuellen privaten Cloud (VPC) wie Ihr Valkey oder Redis OSS auf Ihrer Instanz (empfohlen).
 - Wenn sie unterschiedlich VPCs sind, richten Sie VPC Peering ein, um den Zugriff zwischen den Knoten zu ermöglichen. Weitere Informationen zum VPC Peering finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC](#)
 - Die mit Ihren Valkey- oder OSS Redis-Instances verknüpfte Sicherheitsgruppe sollte eingehenden Datenverkehr von Knoten zulassen. ElastiCache
4. Stellen Sie sicher, dass Ihre Anwendung den Datenverkehr nach Abschluss der Datenmigration an die ElastiCache Knoten weiterleiten kann. Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC](#).

Testen der Datenmigration

Nachdem alle Voraussetzungen erfüllt sind, können Sie die Migrationseinrichtung mit AWS Management Console ElastiCache API, oder überprüfen AWS CLI. Das folgende Beispiel zeigt die Verwendung vonCLI.

Testen Sie die Migration, indem Sie den Befehl `test-migration` mit den folgenden Parametern aufrufen:

- `--replication-group-id` – Die ID der Replikationsgruppe, in die Daten migriert werden sollen.
- `--customer-node-endpoint-list` – Die Liste der Endpunkte, von denen Daten migriert werden sollen. Die Liste sollte nur ein Element enthalten.

Das Folgende ist ein Beispiel mit derCLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache validiert das Migrations-Setup ohne tatsächliche Datenmigration.

Migration starten

Nachdem alle Voraussetzungen erfüllt sind, können Sie mit der Datenmigration beginnen, indem Sie AWS Management Console ElastiCache API, oder verwenden AWS CLI. Bei aktiviertem

Clustermodus und einer unterschiedlichen Slot-Migration wird vor der Live-Migration ein Resharding durchgeführt. Das folgende Beispiel zeigt die Verwendung von CLI.

Note

Wir empfehlen die Verwendung `TestMigration` API zur Validierung des Migrations-Setups. Dies ist allerdings völlig optional.

Starten Sie die Migration, indem Sie den Befehl `start-migration` mit den folgenden Parametern aufrufen:

- `--replication-group-id`— ID der ElastiCache Zielreplikationsgruppe
- `--customer-node-endpoint-list`— Eine Liste von Endpunkten mit IP-Adressen und dem Port, auf dem Ihr Valkey- oder OSS Redis-Quellcluster läuft. DNS Die Liste kann sowohl bei deaktiviertem Clustermodus als auch bei aktiviertem Clustermodus nur ein Element enthalten. Wenn Sie die verkettete Replikation aktiviert haben, kann der Endpunkt auf ein Replikat statt auf den primären Knoten in Ihrem Valkey- oder Redis-Cluster verweisen. OSS

Im Folgenden finden Sie ein Beispiel mit der CLI

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

Während Sie diesen Befehl ausführen, konfiguriert sich der ElastiCache primäre Knoten (in jedem Shard) so, dass er ein Replikat Ihrer Valkey- oder OSS Redis-Instanz wird (in einem entsprechenden Shard, der dieselben Slots in einem clusterfähigen Redis besitzt). Der Status des ElastiCache Clusters ändert sich in „Migrieren“ und die Datenmigration von Ihrer Valkey- oder Redis-Instanz zum Primärknoten beginnt. OSS ElastiCache Abhängig von der Größe der Daten und der Auslastung Ihrer Valkey- oder OSS Redis-Instanz kann es eine Weile dauern, bis die Migration abgeschlossen ist. Sie können den Fortschritt der Migration überprüfen, indem Sie den INFO Befehl [valkey-cli auf Ihrer Valkey-Instanz](#) und Ihrem Primärknoten ausführen. ElastiCache

Nach erfolgreicher Replikation werden alle Schreibvorgänge auf Ihre Valkey- oder OSS Redis-Instanzen auf den Cluster übertragen. ElastiCache Sie können ElastiCache Knoten für Lesevorgänge verwenden. Sie können jedoch nicht in den ElastiCache Cluster schreiben. Wenn mit einem ElastiCache Primärknoten andere Replikatknoten verbunden sind, replizieren diese Replikatknoten

weiterhin vom ElastiCache Primärknoten aus. Auf diese Weise werden alle Daten aus Ihrem Valkey- oder OSS Redis-Cluster auf alle Knoten im Cluster repliziert. ElastiCache

Wenn ein ElastiCache primärer Knoten kein Replikat Ihrer Valkey- oder OSS Redis-Instanz werden kann, versucht er es mehrmals, bevor er sich schließlich wieder zum primären Knoten hochstufte. Der Status des ElastiCache Clusters ändert sich dann in „Verfügbar“, und es wird ein Ereignis der Replikationsgruppe gesendet, dass die Migration nicht initiiert werden konnte. Prüfen Sie Folgendes, um einen solchen Fehlschlag zu diagnostizieren:

- Betrachten Sie das Replikationsgruppenereignis. Verwenden Sie die spezifische Informationen aus dem Ereignis, um den Migrationsfehler zu beheben.
- Wenn das Ereignis keine spezifischen Informationen bereitstellt, müssen Sie sicherstellen, dass die Leitlinien in [Vorbereiten Ihrer Quelle und Ihres Ziels für die Migration](#) befolgt wurden.
- Stellen Sie sicher, dass die Routing-Konfiguration für Ihre VPC und Subnetze den Datenverkehr zwischen ElastiCache Knoten und Ihren Valkey- oder OSS Redis-Instances zulässt.
- Stellen Sie sicher, dass die mit Ihren Valkey- oder OSS Redis-Instances verknüpfte Sicherheitsgruppe eingehenden Datenverkehr von Knoten zulässt. ElastiCache
- Weitere Informationen zu replikationsspezifischen Fehlern finden Sie in den Valkey- oder OSS Redis-Protokollen für Ihre Instances.

Verifizieren des Fortschritts der Datenmigration

Nach Beginn der Datenmigration können Sie den Verlauf folgendermaßen überwachen:

- Stellen Sie sicher, dass Valkey oder Redis OSS `master_link_status` up im INFO Befehl auf den ElastiCache primären Knoten enthalten ist. Sie finden diese Informationen auch in der ElastiCache Konsole. Wählen Sie den Cluster aus und beobachten Sie unter CloudWatch Metriken den Integritätsstatus des primären Links. Die Daten sind synchronisiert, sobald der Wert 1 erreicht hat.
- Sie können überprüfen, ob das ElastiCache Replikat online ist, indem Sie den INFO Befehl auf Ihren Valkey- oder OSS Redis-Instances ausführen. Dadurch werden auch Informationen zur Replikationsverzögerung bereitgestellt.
- Überprüfen Sie, ob der Client-Ausgabepuffer niedrig ist, indem Sie den [CLIENTLIST](#) Befehl auf Ihren Valkey- oder Redis-Instances verwenden. OSS

Nach Abschluss der Datenmigration sind die Daten mit allen neuen Schreibvorgängen auf die Primärknoten Ihres Valkey- oder Redis-Clusters synchronisiert. OSS

Datenmigration abschließen

Wenn Sie bereit sind, zum ElastiCache Cluster zu wechseln, verwenden Sie den `complete-migration` CLI Befehl mit den folgenden Parametern:

- `--replication-group-id` – Die ID für die Replikationsgruppe.
- `--force` – Ein Wert, der zwingt, dass die Migration beendet wird, ohne sicherzustellen, dass die Daten synchron sind.

Im Folgenden wird ein Beispiel gezeigt.

```
aws elasticache complete-migration --replication-group-id test-cluster
```

Während Sie diesen Befehl ausführen, beendet der ElastiCache primäre Knoten (in jedem Shard) die Replikation von Ihrer Valkey- oder OSS Redis-Instance und stuft ihn zum primären Knoten herauf. Diese Übergabe wird normalerweise innerhalb von Minuten abgeschlossen. Prüfen Sie zum Bestätigen der Übergabe an den primären Knoten auf das Ereignis `Complete Migration successful for test-cluster`. An diesem Punkt können Sie Ihre Anwendung auf Schreib- und Lesevorgänge ausrichten. ElastiCache ElastiCache Der Clusterstatus sollte von `migriert` zu `verfügbar` geändert werden.

Wenn die Heraufstufung zum Primärknoten fehlschlägt, repliziert der ElastiCache primäre Knoten weiterhin von Ihrer Valkey- oder Redis-Instanz aus. OSS Der ElastiCache Cluster befindet sich weiterhin im Migrationsstatus, und es wird eine Ereignismeldung über den Fehler an die Replikationsgruppe gesendet. Beachten Sie zum Beheben dieses Fehlers Folgendes:

- Prüfen Sie das Replikationsgruppenereignis. Verwenden Sie spezifische Informationen aus dem Ereignis, um den Fehler zu beheben.
- Unter Umständen wird eine Ereignismeldung angezeigt, weil die Daten nicht synchronisiert sind. Wenn ja, stellen Sie sicher, dass die ElastiCache primäre Instanz von Ihrer Valkey- oder OSS Redis-Instanz aus replizieren kann und beide synchron sind. Wenn Sie die Migration weiterhin stoppen möchten, können Sie den vorhergehenden Befehl mit der Option `--force` ausführen.

- Möglicherweise erhalten Sie eine Ereignismeldung, wenn einer der ElastiCache Knoten ausgetauscht wird. Sie können den Schritt zum Abschließen der Migration wiederholen, nachdem der Austausch abgeschlossen wurde.

Durchführen einer Online-Datenmigration mithilfe der Konsole

Sie können den verwenden AWS Management Console , um Ihre Daten von Ihrem Cluster auf Ihren Valkey- oder OSS Redis-Cluster zu migrieren.

So führen Sie eine Online-Datenmigration mit der Konsole durch

1. Melden Sie sich bei der Konsole an und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Erstellen Sie entweder einen neuen Valkey- oder OSS Redis-Cluster oder wählen Sie einen vorhandenen Cluster aus. Stellen Sie sicher, dass der Cluster die folgenden Anforderungen erfüllt:
 - Ihre Engine-Version sollte Valkey 7.2 und höher oder Redis OSS 5.0.6 oder höher sein.
 - Ihr Cluster sollte nicht aktiviert worden sein. AUTH
 - Die Konfiguration `protected-mode` sollte auf `eingestellt` sein.
 - Wenn Sie eine `bind` Konfiguration in Ihrer Valkey- oder OSS Redis-Konfiguration haben, sollte diese aktualisiert werden, um Anfragen von ElastiCache Knoten zuzulassen.
 - Die Anzahl der Datenbanken sollte zwischen dem ElastiCache Knoten und Ihrem Valkey- oder Redis-Cluster gleich sein. OSS Dieser Wert wird `databases` in der Engine-Konfiguration festgelegt.
 - Valkey- oder OSS Redis-Befehle, die Datenänderungen vornehmen, sollten nicht umbenannt werden, damit die Daten erfolgreich repliziert werden können.
 - Um die Daten aus Ihrem Valkey- oder OSS Redis-Cluster zu replizieren ElastiCache, stellen Sie sicher, dass ausreichend Speicherplatz vorhanden ist, um diese zusätzliche CPU Last zu bewältigen. Diese Last stammt aus der RDB Datei, die von Ihrem Valkey- oder OSS Redis-Cluster erstellt und über das Netzwerk an den Knoten übertragen wurde. ElastiCache
 - Der Cluster hat den Status `available` (Verfügbar).
3. Wählen Sie bei ausgewähltem Cluster Migrieren von Daten von Endpoint für Aktionen aus.
4. Geben Sie im Dialogfeld Daten vom Endpunkt migrieren die IP-Adresse und den Port ein, an dem Ihr Valkey- oder OSS Redis-Cluster verfügbar ist.

⚠ Important

Die IP-Adresse muss exakt übereinstimmen. Wenn Sie die Adresse falsch eingeben, schlägt die Migration fehl.

5. Wählen Sie Start Migration (Migration starten) aus.

Sobald der Cluster die Migration beginnt, wechselt sein Status zu Modifying (Änderung läuft) und dann zu Migrating (Migration läuft).

6. Überwachen Sie den Migrationsfortschritt, indem Sie im Navigationsbereich Events (Ereignisse) auswählen.

Sie können die laufende Migration jederzeit stoppen. Wählen Sie zu diesem Zweck den Cluster und anschließend Stop Data Migration (Datenmigration stoppen) für Actions (Aktionen) aus. Der Cluster wechselt dann zum Status Available (Verfügbar).

Wenn die Migration erfolgreich verläuft, wechselt der Cluster zum Status Available (Verfügbar) und das Ereignisprotokoll enthält folgenden Eintrag:

```
Migration operation succeeded for replication group ElastiCacheClusterName.
```

Wenn die Migration fehlschlägt, wechselt der Cluster zum Status Available (Verfügbar) und das Ereignisprotokoll enthält folgenden Eintrag:

```
Migration operation failed for replication group ElastiCacheClusterName.
```

Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache

Sie können Ihren ElastiCache Clustern zusätzliche Skalierbarkeit und Zuverlässigkeit bieten, indem Sie Regionen und Availability Zones mithilfe des entsprechenden Endpunkts festlegen.

AWS Cloud-Computing-Ressourcen sind in hochverfügbaren Rechenzentrumseinrichtungen untergebracht. Die Einrichtungen dieser Rechenzentren befinden sich an verschiedenen Standorten, um die Skalierbarkeit und Zuverlässigkeit zu erhöhen. Diese Standorte sind nach Regionen und Availability Zones kategorisiert.

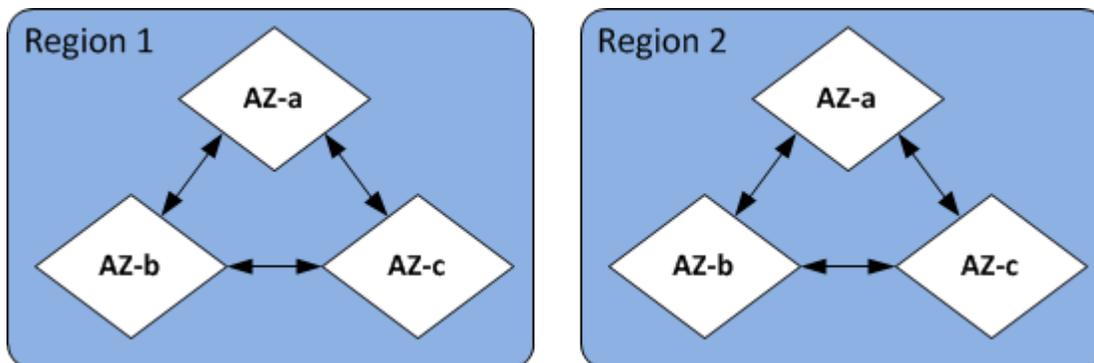
AWS Die Regionen sind groß und weit über verschiedene geografische Standorte verteilt. Availability Zones sind unterschiedliche Standorte innerhalb einer AWS Region, die so konzipiert

sind, dass sie von Ausfällen in anderen Availability Zones isoliert sind. Sie bieten kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben AWS Region.

⚠ Important

Jede Region ist komplett eigenständig. Jede ElastiCache Aktivität, die Sie initiieren (z. B. das Erstellen von Clustern), wird nur in Ihrer aktuellen Standardregion ausgeführt.

Wenn Sie einen Cluster in einer bestimmten Region erstellen oder damit arbeiten möchten, müssen Sie den entsprechenden regionalen Service-Endpunkt wählen. Informationen zu Service-Endpunkten finden Sie unter [Unterstützte Regionen und Endpunkte](#).



Regionen und Availability Zones

Themen

- [Überlegungen zur Availability Zone bei Memcached](#)
- [Lokalisieren Ihrer Knoten](#)
- [Unterstützte Regionen und Endpunkte](#)
- [Verwenden von lokalen Zonen mit ElastiCache](#)
- [Outposts verwenden mit ElastiCache](#)

Überlegungen zur Availability Zone bei Memcached

Die Verteilung Ihrer Memcached-Knoten auf mehrere Availability Zones innerhalb einer Region schützt Sie vor den Auswirkungen eines katastrophalen Ausfalls, z. B. eines Stromausfalls innerhalb einer Availability Zone.

Serverless-Caching

ElastiCache Beim serverlosen Caching wird ein hochverfügbarer Cache erstellt, der sich über mehrere Availability Zones erstreckt. Sie können Subnetze aus verschiedenen Availability Zones angeben, genauso VPC wie Sie Ihren serverlosen Cluster erstellen, oder Sie wählen Subnetze ElastiCache automatisch aus Ihrem Standard aus. VPC

Entwerfen Sie Ihren eigenen ElastiCache (Memcached-) Cluster

Ein Memcached-Cluster kann bis zu 300 Knoten haben. Wenn Sie Knoten erstellen oder zu Ihrem Memcached-Cluster hinzufügen, können Sie eine einzige Availability Zone für all Ihre Knoten angeben, die Auswahl einer einzigen Availability Zone für alle Ihre Knoten zulassen ElastiCache , die Availability Zones für jeden Knoten angeben oder die Auswahl einer Availability Zone für jeden Knoten ermöglichen ElastiCache . Wenn Sie neue Knoten zu einem bestehenden Memcached-Cluster hinzufügen, können diese in verschiedenen Availability Zones erstellt werden. Sobald ein Cache-Knoten erstellt wurde, kann seine Availability Zone nicht mehr geändert werden.

Wenn Sie möchten, dass die Knoten eines Clusters in einem einzelnen Availability Zone-Cluster auf mehrere Availability Zones verteilt sind, ElastiCache können Sie neue Knoten in den verschiedenen Availability Zones erstellen. Anschließend können Sie einige oder alle ursprünglichen Cache-Knoten löschen. Wir empfehlen diesen Ansatz.

Migration von Memcached-Knoten aus einer einzelnen Availability Zone in mehrere Availability Zones

1. Ändern Sie Ihren Cluster, indem Sie neue Cache-Knoten in den Availability Zones erstellen, in denen Sie sie benötigen. Führen Sie in Ihrer Anforderung folgende Schritte aus:
 - Stellen Sie AZMode (CLI:- -az-mode) auf `incross-az`.
 - Setzen Sie NumCacheNodes (CLI:- -num-cache-nodes) auf die Anzahl der derzeit aktiven Cache-Knoten plus die Anzahl der neuen Cache-Knoten, die Sie erstellen möchten.
 - Setzen Sie NewAvailabilityZones (CLI:- -new-availability-zones) auf eine Liste der Zonen, in denen die neuen Cache-Knoten erstellt werden sollen. Um die Availability Zone für jeden neuen Knoten ElastiCache bestimmen zu lassen, geben Sie keine Liste an.
 - Setzen Sie ApplyImmediately (CLI:- -apply-immediately) auf `true`.

 Note

Wenn Sie keine automatische Erkennung verwenden, stellen Sie sicher, dass Sie Ihre Client-Anwendung mit den neuen Cache-Knoten-Endpunkten aktualisieren.

Bevor Sie mit dem nächsten Schritt fortfahren, müssen Sie sicherstellen, dass die Memcached-Knoten vollständig erstellt und verfügbar sind.

2. Ändern Sie den Cluster, indem Sie die Knoten entfernen, die Sie in der ursprünglichen Availability Zone nicht mehr benötigen. Führen Sie in Ihrer Anforderung folgende Schritte aus:
 - Stellen Sie NumCacheNodes (CLI: `- num-cache-nodes`) auf die Anzahl der aktiven Cache-Knoten ein, die Sie benötigen, nachdem diese Änderung angewendet wurde.
 - Setzen Sie CacheNodeIdsToRemove (CLI: `- nodes-to-remove`) auf eine Liste der Cache-Knoten, die Sie aus dem Cluster entfernen möchten.

Die Anzahl der IDs aufgelisteten Cache-Knoten muss der Anzahl der derzeit aktiven Knoten abzüglich des Werts in `entsprechenNumCacheNodes`.

- (Optional) Setzen Sie `ApplyImmediately` (CLI: `- apply-immediately`) auf `true`.

Wenn Sie `ApplyImmediately` (CLI: `- apply-immediately`) nicht auf `true` setzen, werden die Knoten in Ihrem nächsten Wartungsfenster gelöscht.

Lokalisieren Ihrer Knoten

Amazon ElastiCache unterstützt die Lokalisierung aller Knoten eines Clusters in einer oder mehreren Availability Zones (AZs). Wenn Sie sich außerdem dafür entscheiden, Ihre Knoten in mehreren zu lokalisieren AZs (empfohlen ElastiCache), können Sie entweder die AZ für jeden Knoten auswählen oder zulassen ElastiCache, dass sie für Sie ausgewählt werden.

Indem Sie die Knoten an verschiedenen Orten platzieren AZs, verhindern Sie, dass ein Ausfall, z. B. ein Stromausfall, in einer AZ zum Ausfall Ihres gesamten Systems führt. Tests haben gezeigt, dass es keinen signifikanten Latenzunterschied zwischen der Lokalisierung aller Knoten in einer AZ oder deren Verteilung auf mehrere AZs gibt.

Sie können für jeden Knoten eine AZ angeben, wenn Sie einen Cluster erstellen oder beim Ändern eines vorhandenen Clusters Knoten hinzufügen. Weitere Informationen finden Sie hier:

- [Einen Cluster für Memcached erstellen](#)
- [Einen Cluster für Valkey oder Redis erstellen OSS](#)
- [Einen ElastiCache Cluster ändern](#)
- [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)

Unterstützte Regionen und Endpunkte

Amazon ElastiCache ist in mehreren AWS Regionen verfügbar. Das bedeutet, dass Sie ElastiCache Cluster an Standorten starten können, die Ihren Anforderungen entsprechen. Sie können beispielsweise in der AWS Region starten, die Ihren Kunden am nächsten ist, oder in einer bestimmten AWS Region, um bestimmte gesetzliche Anforderungen zu erfüllen.

Jede -Region ist darauf ausgelegt, vollständig von den anderen -Regionen getrennt zu sein. Innerhalb jeder Region gibt es mehrere Availability Zones (AZ). ElastiCache Serverlose Caches replizieren Daten automatisch über mehrere Availability Zones hinweg (außer wenn Daten in zwei Availability Zones repliziert werden) us-west-1, um eine hohe Verfügbarkeit zu gewährleisten. Beim Entwerfen Ihres eigenen ElastiCache Clusters können Sie wählen, ob Sie Ihre Knoten auf unterschiedlichen Wegen starten möchten, um Fehlertoleranz AZs zu erreichen. Weitere Informationen zu Regionen und Availability Zones erhalten Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#) oben in diesem Thema.

Regionen, in denen ElastiCache es unterstützt wird

Regionsname/Region	Endpoint	Protokoll	
Region USA Ost (Ohio) us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS	
Region USA Ost (Nord-Virginia) us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS	
Region US West (N. California) us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS	
Region USA West (Oregon) us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS	
Region Kanada (Zentral) ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS	
Region Kanada (West) ca-west-1	elasticache.ca-west-1.amazonaws.com	HTTPS	
Asien-Pazifik (Jakarta) ap-southeast-3	elasticache.ap-southeast-3.amazonaws.com	HTTPS	
Region Asien-Pazifik (Mumbai)	elasticache.ap-south-1.amazonaws.com	HTTPS	

Regionsname/Region	Endpoint	Protokoll	
ap-south-1			
Region Asien-Pazifik (Hyderabad)	elasticache.ap- south-2.amaz onaws.com	HTTPS	
ap-south-2			
Region Asien-Pazifik (Tokio)	elasticache.ap- northeast-1. amazonaws.com	HTTPS	
ap-northeast-1			
Region Asien-Pazifik (Seoul)	elasticache.ap- northeast-2. amazonaws.com	HTTPS	
ap-northeast-2			
Region Asien-Pazifik (Osaka)	elasticache.ap- northeast-3. amazonaws.com	HTTPS	
ap-northeast-3			
Region Asien-Pazifik (Singapur)	elasticache.ap- southeast-1. amazonaws.com	HTTPS	
ap-southeast-1			
Region Asien-Pazifik (Sydney)	elasticache.ap- southeast-2. amazonaws.com	HTTPS	
ap-southeast-2			
Region Europa (Frankfurt)	elasticache.eu- central-1.am azonaws.com	HTTPS	
eu-central-1			

Regionsname/Region	Endpoint	Protokoll	
Region Europa (Zürich) eu-central-2	elasticache.eu- central-2.am azonaws.com	HTTPS	
Region Europa (Stockholm) eu-north-1	elasticache.eu- north-1.amaz onaws.com	HTTPS	
Region Naher Osten (Bahrain) me-south-1	elasticache.me- south-1.amaz onaws.com	HTTPS	
Region Naher Osten (UAE) me-central-1	elasticache.me- central-1.am azonaws.com	HTTPS	
Region Europa (Irland) eu-west-1	elasticache.eu- west-1.amazo naws.com	HTTPS	
Region Europa (London) eu-west-2	elasticache.eu- west-2.amazo naws.com	HTTPS	
Region Europa (Paris) eu-west-3	elasticache.eu- west-3.amazo naws.com	HTTPS	
Region Europa (Mailand) eu-south-1	elasticache.eu- south-1.amaz onaws.com	HTTPS	

Regionsname/Region	Endpunkt	Protokoll	
Region Europa (Spanien) eu-south-2	elasticache.eu- south-2.amaz onaws.com	HTTPS	
Region Südamerika (São Paulo) sa-east-1	elasticache.sa- east-1.amazo naws.com	HTTPS	
Region China (Peking) cn-north-1	elasticache.cn- north-1.amaz onaws.com.cn	HTTPS	
Region China (Ningxia) cn-northwest-1	elasticache.cn- northwest-1. amazonaws .com.cn	HTTPS	
Region Asien-Pazifik (Hongkong) ap-east-1	elasticache.ap- east-1.amazo naws.com	HTTPS	
Region Afrika (Kapstadt) af-south-1	elasticache.af- south-1.amaz onaws.com	HTTPS	
Region Israel (Tel Aviv) il-central-1	elasticache.il- central-1.am azonaws.com	HTTPS	

Regionsname/Region	Endpoint	Protokoll
AWS GovCloud (US-West) us-gov-west-1	elasticache.us-gov-west-1.amazonaws.com	HTTPS
AWS GovCloud (US-Ost) us-gov-east-1	elasticache.us-gov-east-1.amazonaws.com	HTTPS

Informationen zur Nutzung von AWS GovCloud (US) mit ElastiCache finden Sie unter [Dienste in der Region AWS GovCloud \(USA\): ElastiCache](#).

Einige Regionen unterstützen eine Untermenge von Knotentypen. Eine Tabelle der unterstützten Knotentypen nach AWS Regionen finden Sie unter [Unterstützte Knotentypen nach AWS -Region](#).

Eine Tabelle mit AWS Produkten und Dienstleistungen nach Regionen finden Sie unter [Produkte und Dienstleistungen nach Regionen](#).

Verwenden von lokalen Zonen mit ElastiCache

Eine lokale Zone ist eine Erweiterung einer AWS Region, die sich geografisch in der Nähe Ihrer Benutzer befindet. Sie können jede virtuelle private Cloud (VPC) von einer übergeordneten AWS Region auf eine Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es der lokalen Zone zuweisen. Wenn Sie ein Subnetz in einer lokalen Zone erstellen, VPC wird Ihr Subnetz auf diese lokale Zone erweitert. Das Subnetz in der lokalen Zone funktioniert genauso wie andere Subnetze in Ihrer VPC.

Mithilfe von Local Zones können Sie Ressourcen wie einen ElastiCache Cluster an mehreren Standorten in der Nähe Ihrer Benutzer platzieren.

Wenn Sie einen ElastiCache Cluster erstellen, können Sie ein Subnetz in einer lokalen Zone auswählen. Local Zones haben ihre eigenen Verbindungen mit dem Internet und unterstützen AWS Direct Connect. Daher können Ressourcen, die in einer lokalen Zone erstellt wurden, von lokalen Benutzern mit Kommunikationen mit sehr geringer Latenz genutzt werden. Weitere Informationen finden Sie unter [AWS -Local-Zones](#).

Eine lokale Zone wird beispielsweise `us-west-2-lax-1a` durch einen AWS Regionalcode gefolgt von einer Kennung dargestellt, die den Standort angibt.

Zu diesem Zeitpunkt sind die verfügbaren Local Zones `us-west-2-lax-1a` und `us-west-2-lax-1b`.

Die folgenden Einschränkungen gelten ElastiCache für Local Zones:

- Globale Datenspeicher werden nicht unterstützt.
- Online-Migration wird nicht unterstützt.
- Die folgenden Knotentypen werden von Local Zones zur Zeit unterstützt:

- Aktuelle Generation:

M5-Knotentypen: `cache.m5.large`, `cache.m5.xlarge`, `cache.m5.2xlarge`,
`cache.m5.4xlarge`, `cache.m5.12xlarge`, `cache.m5.24xlarge`

R5-Knotentypen: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`,
`cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

T3-Knotentypen: `cache.t3.micro`, `cache.t3.small`, `cache.t3.medium`

Aktivieren einer Local Zone

1. Aktivieren Sie die lokale Zone in der EC2 Amazon-Konsole.

Weitere Informationen finden Sie unter [Enabling Local Zones](#) im EC2Amazon-Benutzerhandbuch.

2. Erstellen Sie ein Subnetz in der Local Zone.

Weitere Informationen finden Sie unter [Erstellen eines Subnetzes in Ihrem VPC](#) im VPCAmazon-Benutzerhandbuch.

3. Erstellen Sie eine ElastiCache Subnetzgruppe in der lokalen Zone.

Wenn Sie eine ElastiCache Subnetzgruppe erstellen, wählen Sie die Availability Zone-Gruppe für die Lokale Zone aus.

Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

4. Erstellen Sie einen ElastiCache (Memcached-) Cluster, der das ElastiCache Subnetz in der lokalen Zone verwendet.

Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

5. Erstellen Sie einen ElastiCache (Redis-OSS) Cluster, der das ElastiCache Subnetz in der lokalen Zone verwendet. Weitere Informationen finden Sie in den folgenden Themen:

- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Outposts verwenden mit ElastiCache

Sie können AWS Outposts mit ElastiCache verwenden. Outposts ist ein vollständig verwalteter Service, der AWS InfrastrukturAPIs, Dienste und Tools auf Kundenstandorte ausdehnt. Durch den lokalen Zugriff auf die AWS verwaltete Infrastruktur ermöglicht AWS Outposts Kunden, Anwendungen vor Ort mit denselben Programmierschnittstellen wie in AWS Regionen zu erstellen und auszuführen und gleichzeitig lokale Rechen- und Speicherressourcen für geringere Latenz und lokale Datenverarbeitungsanforderungen zu nutzen. Ein Outpost ist ein Pool von AWS Rechen- und Speicherkapazität, der am Standort eines Kunden bereitgestellt wird. AWS betreibt, überwacht und verwaltet diese Kapazität als Teil einer AWS Region. Sie können Subnetze in Ihrem Outpost erstellen und diese angeben, wenn Sie AWS Ressourcen wie ElastiCache Cluster erstellen.

Note

In dieser Version gelten folgende Einschränkungen:

- ElastiCache for Outposts unterstützt nur M5- und R5-Knotenfamilien.
- Multi-AZ (Cross-Outpost-Replikation) wird nicht unterstützt.
- Live-Migration wird nicht unterstützt.
- Lokale Snapshots werden nicht unterstützt.
- Engine-Logs und Slow-Logs können nicht aktiviert werden.
- ElastiCache on Outposts unterstützt CoIP nicht.
- ElastiCache for Outposts wird in den folgenden Regionen nicht unterstützt: cn-north-1, cn-northwest-1 und ap-northeast-3.

Outposts mit der ElastiCache Konsole verwenden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich entweder Valkey-Caches, Redis-Caches oder OSS Memcached-Caches aus.
3. Wenn Sie Valkey-Caches ausgewählt haben, wählen Sie Valkey-Cache erstellen aus. Wenn Sie OSSRedis-Caches ausgewählt haben, wählen Sie Redis-Cache erstellen aus. OSS Wenn Sie Memcached-Caches ausgewählt haben, wählen Sie Memcached-Cache erstellen.
4. Wählen Sie unter Cluster-Einstellungen die Optionen Eigenen Cache entwerfen und Cluster-Cache aus. Lassen Sie den Clustermodus auf Deaktiviert eingestellt. Erstellen Sie dann einen Namen und eine optionale Beschreibung für den Cache.
5. Wählen Sie als Standort die Option Lokal aus.
6. Im Bereich Lokal sehen Sie das Feld Outpost ID. Geben Sie die ID ein, unter der der Cluster ausgeführt werden soll.

Alle weiteren Einstellungen unter Cluster-Einstellungen können die Standardeinstellungen beibehalten.

7. Wählen Sie unter Konnektivität die Option Neue Subnetzgruppe erstellen aus und geben Sie die VPCID ein. Belassen Sie den Rest als Standard und wählen Sie Weiter aus.

Konfiguration von lokalen Optionen

Sie können entweder einen verfügbaren Outpost auswählen, um Ihren Cache-Cluster hinzuzufügen, oder, falls keine Outposts verfügbar sind, einen neuen erstellen, indem Sie die folgenden Schritte ausführen:

Unter Lokale Optionen:

1. Gehen Sie je nach bevorzugter Engine unter OSSValkey-Einstellungen, Redis-Einstellungen oder Memcached-Einstellungen wie folgt vor:
 - a. Name: Geben Sie einen Namen für den Cluster ein
 - b. Beschreibung: Geben Sie eine Beschreibung für den Cluster ein.
 - c. Kompatibilität der Engine-Version: Die Engine-Version basiert auf der AWS Outpost-Region

- d. Port: Akzeptieren Sie für Valkey oder Redis den Standardport OSS 6379. Akzeptieren Sie für Memcached den Standardport 11211. Wenn Sie lieber einen anderen Port verwenden möchten, geben Sie die Portnummer ein.
- e. Parametergruppe: Verwenden Sie das Dropdown-Menü, um eine Standard- oder benutzerdefinierte Parametergruppe auszuwählen.
- f. Knoten-Typ: Die verfügbaren Instances basieren auf der Verfügbarkeit von Outposts. Wenn Sie Valkey oder Redis verwenden OSS, Portierungsassistent für. NETfor Outposts unterstützt nur M5- und R5-Knotenfamilien. Wählen Sie in der Dropdown-Liste die Option Outposts und wählen Sie dann einen verfügbaren Knotentyp aus, den Sie für diesen Cluster verwenden möchten. Wählen Sie dann Speichern aus.
- g. Anzahl der Replikate: Geben Sie die Anzahl der Lesereplikate ein, die für diese Replikationsgruppe erstellt werden sollen. Sie müssen mindestens über ein und nicht mehr als fünf Lesereplikate verfügen. Der Standardwert lautet 2.

Die automatisch generierten Namen der Lesereplikate folgen dem gleichen Muster wie der Name des primären Clusters, wobei ein Bindestrich und eine sequenzielle dreistellige Zahl am Ende hinzugefügt werden, beginnend mit -002. Wenn Ihre Replikationsgruppe beispielsweise den Namen MyGroup hat, dann würden die Namen der Secondaries MyGroup-002, MyGroup-003, MyGroup-004, MyGroup-005, MyGroup-006 lauten.

2. Unter Konnektivität:

- a. Subnetzgruppe: Wählen Sie in der Liste die Option Neu erstellen aus.
 - Name: Geben Sie einen Namen für die Subnetzgruppe ein
 - Beschreibung: Geben Sie eine Beschreibung für die Subnetzgruppe ein
 - VPCID: Die VPC ID sollte mit dem Außenposten VPC übereinstimmen. Wenn Sie in den Outposts eine auswählen VPC, die kein Subnetz IDs hat, wird die Liste leer zurückgegeben.
 - Availability Zone oder Outpost: Wählen Sie den Outpost aus, den Sie verwenden.
 - Subnetz-ID: Wählen Sie eine Subnetz-ID aus, die für den Outpost verfügbar ist. Wenn kein Subnetz IDs verfügbar ist, müssen Sie es erstellen. Weitere Informationen finden Sie unter [Ein Subnetz erstellen](#).
- b. Wählen Sie Erstellen aus.

Anzeige der Details des Outpost-Clusters

Wählen Sie auf der Listenseite einen Cluster aus, der zu einem AWS Outpost gehört, und beachten Sie Folgendes, wenn Sie sich die Cluster-Details ansehen:

- **Availability Zone:** Dies stellt den Außenposten dar, wobei ein ARN (Amazon-Ressourcenname) und die AWS Ressourcennummer verwendet werden.
- **Name des Außenpostens:** Der Name des Außenpostens AWS .

Verwenden von Outposts mit dem AWS CLI

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um mehrere AWS Dienste von der Befehlszeile aus zu steuern und sie mithilfe von Skripten zu automatisieren. Sie können das AWS CLI für (einmalige) Ad-hoc-Operationen verwenden.

Herunterladen und Konfigurieren der AWS CLI

Das AWS CLI läuft unter Windows, MacOS oder Linux. Gehen Sie folgendermaßen vor, um sie herunterzuladen und zu konfigurieren.

Zum Herunterladen, Installieren und Konfigurieren des CLI

1. Laden Sie die Webseite AWS CLI auf der [AWS Befehlszeilenschnittstelle](#) herunter.
2. Folgen Sie den Anweisungen [zur Installation AWS CLI](#) und [Konfiguration von AWS CLI im AWS Command Line Interface](#) Benutzerhandbuch.

AWS CLIMit Outposts verwenden

Verwenden Sie den folgenden CLI Vorgang, um einen Cache-Cluster zu erstellen, der Outposts verwendet:

- [create-cache-cluster](#)— Bei dieser Operation akzeptiert der `outpost-mode` Parameter einen Wert, der angibt, ob die Knoten im Cache-Cluster in einem einzelnen Outpost oder in mehreren Outposts erstellt werden.

Note

Zurzeit wird nur der `single-outpost`-Modus unterstützt.

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id cache cluster id \  
  --outpost-mode single-outpost \  
  \
```

Arbeiten mit ElastiCache

In diesem Abschnitt finden Sie Einzelheiten zur Verwaltung der verschiedenen Komponenten Ihrer ElastiCache Implementierung.

Themen

- [Snapshot und Wiederherstellung](#)
- [Motorversionen und Aufrüstung in ElastiCache](#)
- [ElastiCache bewährte Methoden und Caching-Strategien](#)
- [Verwaltung Ihres selbst entworfenen Clusters in ElastiCache](#)
- [Skalierung ElastiCache](#)
- [Erste Schritte mit JSON für Valkey und Redis OSS](#)
- [Verschlagworten Sie Ihre Ressourcen ElastiCache](#)
- [Verwenden des Amazon ElastiCache Well-Architected-Objektivs](#)
- [Allgemeine Schritte zur Fehlerbehebung und bewährte Methoden mit ElastiCache](#)

Snapshot und Wiederherstellung

ElastiCache Amazon-Caches, auf denen Valkey, Redis oder Serverless Memcached ausgeführt werden, können ihre Daten sichern, indem sie einen Snapshot erstellen. Sie können das Backup verwenden, um einen Cache wiederherzustellen oder Daten in einem neuen Cache durch Seeding zu speichern. Ein Backup besteht aus den Metadaten des Caches zusammen mit allen Daten im Cache. Alle Sicherungen werden in Amazon Simple Storage Service (Amazon S3) geschrieben, der einen dauerhaften Speicher bereitstellt. Sie können Ihre Daten jederzeit wiederherstellen, indem Sie einen neuen Valkey-, Redis- oder Serverless Memcached-Cache erstellen und ihn mit Daten aus einem Backup füllen. Mit ElastiCache können Sie Backups mithilfe von AWS Management Console, AWS CLI und dem AWS Command Line Interface verwalten.

Wenn Sie vorhaben, den Cache zu löschen, und es wichtig ist, die Daten beizubehalten, können Sie eine zusätzliche Vorsichtsmaßnahme ergreifen. Erstellen Sie dazu zuerst ein manuelles Backup, überprüfen Sie, dass der Status `available` lautet und löschen Sie dann den Cache. Dadurch wird sichergestellt, dass die Cache-Daten weiterhin verfügbar sind, falls das Backup fehlschlagen sollte. Sie können nach den oben beschriebenen bewährten Methoden erneut versuchen, eine Sicherung anzufertigen.

Themen

- [Sicherungseinschränkungen](#)
- [Auswirkungen von Backups selbst entworfener Cluster auf die Leistung](#)
- [Planen automatischer Backups](#)
- [Erstellen manueller Backups](#)
- [Erstellen einer endgültigen Sicherung](#)
- [Beschreiben von Sicherungen](#)
- [Kopieren eines Backups](#)
- [Exportieren einer Sicherung](#)
- [Wiederherstellen aus einem Backup in einen neuen Cache](#)
- [Löschen einer Sicherung](#)
- [Markieren von Sicherungen](#)
- [Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup](#)

Sicherungseinschränkungen

Berücksichtigen Sie die folgenden Einschränkungen bei der Planung und Erstellung von Sicherungen:

- Backup und Wiederherstellung werden nur für Caches unterstützt, die auf Valkey, Redis OSS oder Serverless Memcached ausgeführt werden.
- Bei Valkey- oder Redis-Clustern OSS (Clustermodus deaktiviert) werden Sicherung und Wiederherstellung auf Knoten nicht unterstützt. `cache.t1.micro` Alle anderen Cache-Knotentypen werden unterstützt.
- Bei Valkey- oder Redis-Clustern OSS (Clustermodus aktiviert) werden Sicherung und Wiederherstellung für alle Knotentypen unterstützt.
- Während eines zusammenhängenden Zeitraums von 24 Stunden können Sie nicht mehr als 24 manuelle Backups pro serverlosem Cache erstellen. Für OSS selbst entworfene Cluster von Valkey und Redis können Sie nicht mehr als 20 manuelle Backups pro Knoten im Cluster erstellen.
- Valkey oder Redis OSS (Clustermodus aktiviert) unterstützen nur das Erstellen von Backups auf Clusterebene (für die API oder CLI, die Replikationsgruppenebene). Valkey oder Redis OSS (Clustermodus aktiviert) unterstützen nicht das Erstellen von Backups auf Shard-Ebene (für API oder CLI, auf Knotengruppenebene).

- Während des Backup-Vorgangs können Sie keine anderen CLI OD-Operationen im API serverlosen Cache ausführen. Sie können während des API Backups CLI OR-Operationen auf einem selbst entworfenen Cluster ausführen.
- Wenn Sie Valkey- oder OSS Redis-Caches mit Data Tiering verwenden, können Sie kein Backup nach Amazon S3 exportieren.
- Sie können ein Backup eines Clusters mit dem R6gd-Knotentyp nur für Cluster wiederherstellen, die den R6gd-Knotentyp verwenden.

Auswirkungen von Backups selbst entworfener Cluster auf die Leistung

Backups auf Serverless-Caches sind für die Anwendung transparent, ohne dass sich dies auf die Leistung auswirkt. Bei der Erstellung von Backups für selbst entworfene Cluster kann es jedoch je nach verfügbarem reservierten Speicher zu Leistungseinbußen kommen. Backups für selbst entworfene Cluster sind mit ElastiCache (Memcached) nicht verfügbar, aber mit (Redis). ElastiCache OSS

Nachfolgend sind Richtlinien zur Verbesserung der Backup-Leistung für selbst entworfene Cluster aufgeführt.

- `reserved-memory-percent` Parameter festlegen — Um übermäßiges Paging zu vermeiden, empfehlen wir Ihnen, den Parameter festzulegen. `reserved-memory-percent` Dieser Parameter verhindert, dass Valkey und Redis OSS den gesamten verfügbaren Speicher des Knotens verbrauchen, und kann dazu beitragen, den Umfang des Paging zu reduzieren. Die Leistung lässt sich möglicherweise einfach durch Verwenden eines größeren Knotens verbessern. Weitere Hinweise zum reservierten Speicher und zu den Parametern finden Sie unter [reserved-memory-percent](#) [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#)
- Backups aus einer Read Replica erstellen — Wenn Sie Valkey oder Redis OSS in einer Knotengruppe mit mehr als einem Knoten ausführen, können Sie ein Backup vom primären Knoten oder einer der Read Replica erstellen. Aufgrund der dabei benötigten Systemressourcen empfehlen wir BGSAVE, Backups von einer der Read Replicas zu erstellen. Während das Backup aus dem Replikat erstellt wird, bleibt der primäre Knoten von BGSAVE den Ressourcenanforderungen unberührt. Der primäre Knoten kann weiterhin Anfragen verarbeiten, ohne langsamer zu werden.

Siehe dazu [Erstellen einer manuellen Sicherung \(Konsole\)](#) und wählen Sie im Fenster Sicherung erstellen im Feld Clustername ein Replikat anstelle des standardmäßigen Primärknotens aus.

Wenn Sie eine Replikationsgruppe löschen und ein letztes Backup anfordern, wird das Backup ElastiCache immer vom Primärknoten erstellt. Dadurch wird sichergestellt, dass Sie die neuesten Valkey- oder OSS Redis-Daten erfassen, bevor die Replikationsgruppe gelöscht wird.

Planen automatischer Backups

Sie können automatische Backups für jeden OSS serverlosen Valkey- oder Redis-Cache oder jeden selbst entworfenen Cluster aktivieren. Wenn automatische Backups aktiviert sind, ElastiCache erstellt täglich eine Sicherungskopie des Caches. Es gibt keine Auswirkungen auf den Cache und die Änderung erfolgt sofort. Automatische Backups schützen vor Datenverlust. Bei einem Ausfall können Sie einen neuen Cache erstellen, indem Sie Ihre Daten aus dem aktuellen Backup wiederherstellen. Das Ergebnis ist ein warm gestarteter Cache, auf dem Ihre Daten vorab geladen und einsatzbereit sind. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Sie können automatische Backups für jeden Memcached Serverless Cache aktivieren. Wenn automatische Backups aktiviert sind, ElastiCache erstellt täglich eine Sicherungskopie des Caches. Es gibt keine Auswirkungen auf den Cache und die Änderung erfolgt sofort. Automatische Backups schützen vor Datenverlust. Bei einem Ausfall können Sie einen neuen Cache erstellen, indem Sie Ihre Daten aus dem aktuellen Backup wiederherstellen. Das Ergebnis ist ein warm gestarteter Cache, auf dem Ihre Daten vorab geladen und einsatzbereit sind. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Wenn Sie automatische Sicherungen planen, sollten Sie die folgenden Einstellungen vornehmen:

- **Backup-Startzeit** — Die Tageszeit, zu der mit der Erstellung eines Backups ElastiCache begonnen wird. Sie können das Backup-Fenster für jeden beliebigen Zeitpunkt festlegen, zu dem es am sinnvollsten ist. Wenn Sie kein Backup-Fenster angeben, wird automatisch eines ElastiCache zugewiesen.
- **Aufbewahrungsfrist für Sicherungen** – Die Anzahl von Tagen, für die die Sicherung in Amazon S3 aufbewahrt wird. Wenn Sie z. B. als Aufbewahrungsfrist 5 festlegen, dann wird die heute erstellte Sicherung 5 Tage lang aufbewahrt. Bei Ablauf der Aufbewahrungsfrist wird die Sicherung automatisch gelöscht.

Der maximale Aufbewahrungsfrist für Sicherungen ist 35 Tage. Wenn als Backup-Aufbewahrungsfrist „0“ festgelegt wird, werden automatische Backups für den Cache deaktiviert.

Wenn Sie automatische Backups planen, ElastiCache wird mit der Erstellung des Backups begonnen. Sie können das Backup-Fenster für jeden beliebigen Zeitpunkt festlegen, zu dem es am sinnvollsten ist. Wenn Sie kein Backup-Fenster angeben, wird automatisch eines ElastiCache zugewiesen.

Sie können automatische Backups aktivieren oder deaktivieren, wenn Sie entweder einen neuen Cache erstellen oder einen vorhandenen Cache aktualisieren, indem Sie die ElastiCache Konsole, die AWS CLI, die ElastiCache API oder die ElastiCache API verwenden. Bei Valkey und Redis erfolgt dies, indem Sie im Abschnitt Erweiterte Valkey-Einstellungen oder Erweiterte OSS Redis-Einstellungen das Kontrollkästchen Automatische Backups aktivieren aktivieren aktivieren. Für Memcached erfolgt dies, indem Sie im Abschnitt Erweiterte Memcached-Einstellungen das Kontrollkästchen Automatische Backups aktivieren aktivieren aktivieren.

Erstellen manueller Backups

Zusätzlich zu automatischen Sicherungen können Sie jederzeit eine manuelle Sicherung erstellen. Im Gegensatz zu automatischen Sicherungen, die nach einem angegebenen Aufbewahrungszeitraum automatisch gelöscht werden, ist für manuelle Sicherungen kein Aufbewahrungszeitraum vorhanden, nachdem sie automatisch gelöscht werden. Selbst wenn Sie den Cache löschen, werden alle manuellen Backups aus diesem Cache beibehalten. Wenn Sie eine manuelle Sicherung nicht mehr aufbewahren möchten, müssen Sie sie selbst explizit löschen.

Eine manuelle Sicherung kann nicht nur direkt, sondern auch auf eine der folgenden Arten erstellt werden:

- [Kopieren eines Backups](#). Es spielt keine Rolle, ob die Quellsicherung automatisch oder manuell erstellt wurde.
- [Erstellen einer endgültigen Sicherung](#). Erstellen Sie ein Backup unmittelbar vor dem Löschen eines Clusters oder Knotens.

Sie können eine manuelle Sicherung eines Caches mit dem AWS Management Console AWS CLI, dem oder dem erstellen ElastiCache API.

Erstellen einer manuellen Sicherung (Konsole)

So erstellen Sie ein manuelles Backup eines Caches (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich je nach Ihren Vorlieben Valkey-Caches, OSSRedis-Caches oder Memcached-Caches aus.
3. Wählen Sie das Feld links neben dem Namen des Caches aus, den Sie sichern möchten.
4. Wählen Sie Backup.
5. Geben Sie im Dialog Create Backup einen Namen für Ihre Sicherung im Feld Backup Name ein. Es ist empfehlenswert, dass aus dem Namen der gesicherte Cluster und das Datum sowie die Uhrzeit der Sicherung ersichtlich ist.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.

- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

6. Wählen Sie Create Bucket.

Der Status des Clusters ändert sich in snapshotting.

Erstellen einer manuellen Sicherung (AWS CLI)

Manuelles Backup eines serverlosen Caches mit dem AWS CLI

Um eine manuelle Sicherung eines Caches mit dem zu erstellen AWS CLI, verwenden Sie den `create-serverless-snapshot` AWS CLI Vorgang mit den folgenden Parametern:

- `--serverless-cache-name` – der Name des Serverless-Caches, für den Sie ein Backup erstellen.
- `--serverless-cache-snapshot-name` – der Name des zu erstellenden Snapshots.

Für Linux, macOS oder Unix:

- ```
aws elasticache create-serverless-snapshot \
 --serverless-cache-name CacheName \
 --serverless-cache-snapshot-name bkup-20231127
```

Für Windows:

- ```
aws elasticache create-serverless-snapshot ^  
    --serverless-cache-name CacheName ^  
    --serverless-cache-snapshot-name bkup-20231127
```

Manuelles Backup eines selbst entworfenen Clusters mit dem AWS CLI

Um ein manuelles Backup eines selbst entworfenen Clusters mithilfe von zu erstellen AWS CLI, verwenden Sie den `create-snapshot` AWS CLI Vorgang mit den folgenden Parametern:

- `--cache-cluster-id`

- Wenn der Cluster, den Sie sichern, keine Replikatknoten hat, `--cache-cluster-id` ist dies beispielsweise der Name des Clusters, den Sie sichern *mycluster*.
- Wenn der von Ihnen gesicherte Cluster über mindestens einen Replikatknoten verfügt, lautet der Name des Knotens im Clusters, den Sie für die Sicherung verwenden möchten `--cache-cluster-id`. Der Name könnte beispielsweise lauten *mycluster-002*.

Verwenden Sie diesen Parameter nur, wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) sichern.

- `--replication-group-id`— Name des Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) (CLI/API: eine Replikationsgruppe), der als Quelle für das Backup verwendet werden soll. Verwenden Sie diesen Parameter, wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) sichern.
- `--snapshot-name` Name des zu erstellenden Snapshots.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

Beispiel 1: Sicherung eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) ohne Replikatknoten

Mit dem folgenden AWS CLI Vorgang wird das Backup `bkup-20150515` aus dem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) erstellt `myNonClusteredRedis`, der keine Read Replicas enthält.

Für Linux, macOS oder Unix:

```
aws elasticache create-snapshot \  
  --cache-cluster-id myNonClusteredRedis \  
  --snapshot-name bkup-20150515
```

Für Windows:

```
aws elasticache create-snapshot ^
  --cache-cluster-id myNonClusteredRedis ^
  --snapshot-name bkup-20150515
```

Beispiel 2: Sicherung eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mit Replikatknoten

Mit dem folgenden AWS CLI Vorgang wird das Backup `bkup-20150515` aus dem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) erstellt. `myNonClusteredRedis` Diese Sicherung hat eine oder mehrere Lesereplikate.

Für Linux, macOS oder Unix:

```
aws elasticache create-snapshot \
  --cache-cluster-id myNonClusteredRedis-001 \
  --snapshot-name bkup-20150515
```

Für Windows:

```
aws elasticache create-snapshot ^
  --cache-cluster-id myNonClusteredRedis-001 ^
  --snapshot-name bkup-20150515
```

Beispielausgabe: Sicherung eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mit Replikatknoten

Die Ausgabe der Operation sieht in etwa folgendermaßen aus.

```
{
  "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"
  }
}

```

Beispiel 3: Sicherung eines Clusters für Valkey oder Redis OSS (Clustermodus aktiviert)

Der folgende AWS CLI Vorgang erstellt das Backup `bkup-20150515` aus dem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert). `myClusteredRedis` Beachten Sie die Verwendung von `--replication-group-id` anstelle von `--cache-cluster-id` zum Identifizieren der Quelle.

Für Linux, macOS oder Unix:

```

aws elasticache create-snapshot \
  --replication-group-id myClusteredRedis \
  --snapshot-name bkup-20150515

```

Für Windows:

```

aws elasticache create-snapshot ^
  --replication-group-id myClusteredRedis ^
  --snapshot-name bkup-20150515

```

Beispielausgabe: Sicherung eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert)

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus.

```
{
  "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"
  }
}
```

Verwandte Themen

Weitere Informationen finden Sie unter [create-bucket](#) in der AWS CLI -Befehlsreferenz.

Erstellen einer endgültigen Sicherung

Sie können ein endgültiges Backup mit der ElastiCache Konsole AWS CLI, dem oder dem erstellen ElastiCache API.

Erstellen einer endgültigen Sicherung (Konsole)

Sie können eine endgültige Sicherung erstellen, wenn Sie einen serverlosen Valkey- oder Redis-Cache, einen OSS selbst entworfenen Valkey- oder Redis-Cluster oder einen OSS serverlosen Memcached-Cache mithilfe der Konsole löschen. ElastiCache

Um beim Löschen eines Caches eine endgültige Sicherung zu erstellen, wählen Sie im Löschdialogfeld unter Sicherung erstellen die Option Ja aus und geben Sie der Sicherung einen Namen.

Verwandte Themen

- [Unter Verwendung der AWS Management Console](#)
- [Löschen einer Replikationsgruppe \(Konsole\)](#)

Erstellen einer endgültigen Sicherung (AWS CLI)

Sie können ein endgültiges Backup erstellen, wenn Sie einen Cache löschen, indem Sie den AWS CLI

Themen

- [Beim Löschen eines Valkey-Caches, Redis-Caches oder OSS Memcached-Caches ohne Server](#)
- [Beim Löschen eines Valkey- oder OSS Redis-Clusters ohne Read Replicas](#)
- [Beim Löschen eines Valkey- oder OSS Redis-Clusters mit Read Replicas](#)

Beim Löschen eines Valkey-Caches, Redis-Caches oder OSS Memcached-Caches ohne Server

Verwenden Sie den `delete-serverless-cache` AWS CLI Vorgang mit den folgenden Parametern, um ein endgültiges Backup zu erstellen.

- `--serverless-cache-name` – Name des zu löschenden Clusters.
- `--final-snapshot-name` – Name des Backups.

Der folgende Code erstellt das endgültige Backup `bkup-20231127-final` beim Löschen des Caches `myserverlesscache`.

Für Linux, macOS oder Unix:

```
aws elasticache delete-serverless-cache \  
    --serverless-cache-name myserverlesscache \  
    --final-snapshot-name bkup-20231127-final
```

Für Windows:

```
aws elasticache delete-serverless-cache ^  
    --serverless-cache-name myserverlesscache ^  
    --final-snapshot-name bkup-20231127-final
```

Weitere Informationen finden Sie [delete-serverless-cache](#) in der AWS CLI Befehlsreferenz.

Beim Löschen eines Valkey- oder OSS Redis-Clusters ohne Read Replicas

Verwenden Sie den `delete-cache-cluster` AWS CLI Vorgang mit den folgenden Parametern, um ein endgültiges Backup für einen selbst entworfenen Cluster ohne Read Replicas zu erstellen.

- `--cache-cluster-id` Name des zu löschenden Clusters.
- `--final-snapshot-identifizier` Name der Sicherung.

Der folgende Code erstellt die endgültige Sicherung `bkup-20150515-final` beim Löschen des Clusters `myRedisCluster`.

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-cluster \  
    --cache-cluster-id myRedisCluster \  
    --final-snapshot-identifizier bkup-20150515-final
```

Für Windows:

```
aws elasticache delete-cache-cluster ^  
    --cache-cluster-id myRedisCluster ^  
    --final-snapshot-identifizier bkup-20150515-final
```

Weitere Informationen finden Sie [delete-cache-cluster](#) in der AWS CLI Befehlsreferenz.

Beim Löschen eines Valkey- oder OSS Redis-Clusters mit Read Replicas

Verwenden Sie den `delete-replication-group` AWS CLI Vorgang mit den folgenden Parametern, um beim Löschen einer Replikationsgruppe ein letztes Backup zu erstellen:

- `--replication-group-id` Name der zu löschenden Replikationsgruppe.
- `--final-snapshot-identifizier` Name der endgültigen Sicherung.

Der folgende Code erstellt die endgültige Sicherung `bkup-20150515-final` beim Löschen der Replikationsgruppe `myReplGroup`.

Für Linux, macOS oder Unix:

```
aws elasticache delete-replication-group \  
  --replication-group-id myReplGroup \  
  --final-snapshot-identifizier bkup-20150515-final
```

Für Windows:

```
aws elasticache delete-replication-group ^  
  --replication-group-id myReplGroup ^  
  --final-snapshot-identifizier bkup-20150515-final
```

Weitere Informationen finden Sie [delete-replication-group](#) in der AWS CLI Befehlsreferenz.

Beschreiben von Sicherungen

Die folgenden Verfahren zeigen, wie Sie eine Liste Ihrer Sicherungen anzeigen. Wenn Sie möchten, können Sie auch die Details zu einer bestimmten Sicherung anzeigen.

Beschreibung von Backups (Konsole)

Um Backups anzuzeigen, verwenden Sie AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Backups aus.
3. Um die Details zu einer bestimmten Sicherung anzuzeigen, aktivieren Sie das Kontrollkästchen links neben dem Namen der betreffenden Sicherung.

Beschreiben von Serverless-Backups (AWS CLI)

Verwenden Sie den `describe-serverless-cache-snapshots` CLI Vorgang, um eine Liste von serverlosen Backups und optional Details zu einem bestimmten Backup anzuzeigen.

Beispiele

Die folgende Operation listet mit dem Parameter `--max-records` bis zu 20 zum Konto gehörende Sicherungen auf. Wenn der Parameter `--max-records` weggelassen wird, werden bis zu 50 Sicherungen aufgelistet.

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

Die folgende Operation listet mit dem Parameter `--serverless-cache-name` nur die zum Cache `my-cache` gehörenden Backups auf.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

Die folgende Operation listet mit dem Parameter `--serverless-cache-snapshot-name` die Details der Sicherung `my-backup` auf.

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-name my-backup
```

Weitere Informationen finden Sie [describe-serverless-cache-snapshots](#) in der AWS CLI Befehlsreferenz.

Beschreiben von selbst entworfenen Cluster-Backups (AWS CLI)

Verwenden Sie den `describe-snapshots` CLI Vorgang, um eine Liste selbst entworfener Cluster-Backups und optional Details zu einem bestimmten Backup anzuzeigen.

Beispiele

Die folgende Operation listet mit dem Parameter `--max-records` bis zu 20 zum Konto gehörende Sicherungen auf. Wenn der Parameter `--max-records` weggelassen wird, werden bis zu 50 Sicherungen aufgelistet.

```
aws elasticache describe-snapshots --max-records 20
```

Die folgende Operation listet mit dem Parameter `--cache-cluster-id` nur die zum Cluster `my-cluster` gehörenden Sicherungen auf.

```
aws elasticache describe-snapshots --cache-cluster-id my-cluster
```

Die folgende Operation listet mit dem Parameter `--snapshot-name` die Details der Sicherung `my-backup` auf.

```
aws elasticache describe-snapshots --snapshot-name my-backup
```

Weitere Informationen finden Sie unter [describe-snapshots](#) in der Befehlsreferenz. AWS CLI

Kopieren eines Backups

Sie können eine Kopie jedes beliebigen Backups erstellen. Dabei spielt es keine Rolle, ob es automatisch oder manuell erstellt wurde. Sie können Ihr Backup auch exportieren, sodass Sie von außen darauf zugreifen können ElastiCache. Eine Anleitung zum Exportieren Ihrer Sicherung finden Sie unter [Exportieren einer Sicherung](#).

Das folgende Verfahren zeigt, wie Sie ein Backup kopieren.

Kopieren von Backups (Konsole)

So kopieren Sie ein Backup (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste Ihrer Sicherungen anzuzeigen, wählen Sie im linken Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, die Sie kopieren möchten.
4. Wählen Sie Aktionen und dann Kopieren aus.
5. Geben Sie in das Feld New backup name einen Namen für die neue Sicherung ein.
6. Wählen Sie die Option Kopieren aus.

Kopieren eines Serverless-Backups (AWS CLI)

Verwenden Sie die Operation `copy-serverless-cache-snapshot`, um ein Backup eines Serverless-Caches zu kopieren.

Parameter

- `--source-serverless-cache-snapshot-name` Name der zu kopierenden Sicherung.
- `--target-serverless-cache-snapshot-name` Name der Sicherungskopie.

Das folgende Beispiel erstellt eine Kopie einer automatischen Sicherung.

Für Linux, macOS oder 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 my-backup-copy
```

Für 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
```

Weitere Informationen finden Sie unter [copy-serverless-cache-snapshot](#) im AWS CLI.

Kopieren eines selbst entworfenen Cluster-Backups (AWS CLI)

Verwenden Sie die Operation `copy-snapshot`, um ein Backup eines selbst entworfenen Clusters zu kopieren.

Parameter

- `--source-snapshot-name` Name der zu kopierenden Sicherung.
- `--target-snapshot-name` Name der Sicherungskopie.
- `--target-bucket` Reserviert zum Exportieren einer Sicherung. Verwenden Sie den Parameter nicht, um eine Kopie einer Sicherung anzufertigen. Weitere Informationen finden Sie unter [Exportieren einer Sicherung](#).

Das folgende Beispiel erstellt eine Kopie einer automatischen Sicherung.

Für Linux, macOS oder Unix:

```
aws elasticache copy-snapshot \  
--source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 \  
--target-snapshot-name my-backup-copy
```

Für Windows:

```
aws elasticache copy-snapshot ^  
--source-snapshot-name automatic.my-redis-primary-2014-03-27-03-15 ^  
--target-snapshot-name my-backup-copy
```

Weitere Informationen finden Sie unter [copy-snapshot](#) im AWS CLI.

Exportieren einer Sicherung

Amazon ElastiCache unterstützt den Export Ihres ElastiCache (Redis-OSS) Backups in einen Amazon Simple Storage Service (Amazon S3) -Bucket, sodass Sie von außerhalb ElastiCache darauf zugreifen können. Sie können ein Backup mit der ElastiCache Konsole AWS CLI, dem oder dem ElastiCache API exportieren.

Das Exportieren eines Backups kann hilfreich sein, wenn Sie einen Cluster in einer anderen AWS Region starten müssen. Sie können Ihre Daten in eine AWS Region exportieren, die RDB-Datei in die neue AWS Region kopieren und dann diese RDB-Datei verwenden, um den neuen Cache zu speichern, anstatt darauf zu warten, dass der neue Cluster durch Use gefüllt wird. Informationen zum Erstellen eines neuen Clusters durch Seeding finden Sie unter [Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup](#). Ein weiterer Grund, warum Sie die Daten Ihres Caches exportieren möchten, ist die Verwendung der RDB-Datei für die Offline-Verarbeitung.

Important

- Das ElastiCache Backup und der Amazon S3 S3-Bucket, in den Sie es kopieren möchten, müssen sich in derselben AWS Region befinden.

Obwohl Sicherungen, die in einen Amazon-S3-Bucket kopiert werden, verschlüsselt sind, empfehlen wir dringend, dass Sie anderen Personen keinen Zugriff auf den Amazon-S3-Bucket mit Ihren gespeicherten Sicherungen gewähren.

- Das Exportieren eines Backups nach Amazon S3 wird für Cluster, die Daten-Tiering verwenden, nicht unterstützt. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).
- Das Exportieren eines Backups ist für OSS selbst entworfene Valkey- und Redis-Cluster, Serverless Valkey und OSS Redis sowie Serverless Memcached verfügbar. Das Exportieren eines Backups ist für selbst entworfene Memcached-Cluster nicht verfügbar.

Bevor Sie ein Backup in einen Amazon S3 S3-Bucket exportieren können, müssen Sie über einen Amazon S3 S3-Bucket in derselben AWS Region wie das Backup verfügen. Gewähren ElastiCache Sie Zugriff auf den Bucket. Die ersten beiden Schritte zeigen, wie Sie dabei vorgehen.

Erstellen eines Amazon-S3-Buckets

Die folgenden Schritte verwenden die Amazon S3 S3-Konsole, um einen Amazon S3 S3-Bucket zu erstellen, in den Sie Ihr ElastiCache Backup exportieren und speichern.

So erstellen Sie einen Amazon-S3-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Create Bucket (Bucket erstellen) aus.
3. Verfahren Sie unter Create a Bucket – Select a Bucket Name and Region wie folgt:

- a. Geben Sie für Bucket-Name einen Namen für Ihren Amazon-S3-Bucket ein.

Der Name Ihres Amazon S3 S3-Buckets muss DNS -konform sein. Andernfalls ElastiCache kann nicht auf Ihre Backup-Datei zugreifen. Die Regeln für die DNS Einhaltung der Vorschriften lauten wie folgt:

- Bucket-Namen müssen mindestens 3 und dürfen höchstens 63 Zeichen umfassen.
 - Die Namen müssen eine Folge aus einer oder mehreren Beschriftungen darstellen, die durch einen Punkt (.) getrennt sind, wobei jede Beschriftung:
 - mit einem Kleinbuchstaben oder einer Zahl beginnen.
 - mit einem Kleinbuchstaben oder einer Zahl beginnen.
 - Enthält nur Kleinbuchstaben, Zahlen und Bindestriche.
 - Er darf nicht als IP-Adresse (z. B. 192.0.2.0) formatiert sein.
- b. Wählen Sie aus der Regionsliste eine AWS Region für Ihren Amazon S3 S3-Bucket aus. Diese AWS Region muss dieselbe AWS Region sein wie das ElastiCache Backup, das Sie exportieren möchten.
 - c. Wählen Sie Create (Erstellen) aus.

Weitere Informationen zum Erstellen eines Amazon-S3-Buckets finden Sie unter [Erstellen von Buckets](#) im Handbuch für Amazon Simple Storage Service.

Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket

ElastiCache Um einen Snapshot in einen Amazon S3 S3-Bucket kopieren zu können, müssen Sie Ihre Bucket-Richtlinie aktualisieren, um ElastiCache Zugriff auf den Bucket zu gewähren.

⚠ Warning

Obwohl Sicherungen, die in einen Amazon-S3-Bucket kopiert werden, verschlüsselt sind, kann jede Person mit Zugriff auf Ihren Amazon-S3-Bucket auf Ihre Daten zugreifen. Wir empfehlen Ihnen daher dringend, IAM Richtlinien einzurichten, um unbefugten Zugriff auf diesen Amazon S3 S3-Bucket zu verhindern. Weitere Informationen finden Sie unter [Verwalten des Zugriffs](#) im Benutzerhandbuch zu Amazon S3.

Führen Sie die folgenden Schritte aus, um die korrekten Berechtigungen für einen Amazon S3 Bucket zu erstellen.

Um ElastiCache Zugriff auf einen S3-Bucket zu gewähren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Amazon-S3-Buckets aus, in den Sie die Sicherung kopieren möchten. Dies sollte der in [Erstellen eines Amazon-S3-Buckets](#) erstellte S3-Bucket sein.
3. Wählen Sie den Tab Berechtigungen und wählen Sie unter Berechtigungen die Option Zugriffskontrollliste (ACL) und dann Bearbeiten aus.
4. Fügen Sie die Fügen Sie die Kanonische Empfänger-ID
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353 mit den folgenden Optionen hinzu:
 - Objekte – Auflisten oder Schreiben
 - BucketACL: Lesen, Schreiben

i Note

- Für die PDT GovCloud Region lautet die kanonische ID.
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Für die OSU GovCloud Region lautet die kanonische ID.
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c

5. Wählen Sie Save (Speichern) aus.

Exportieren Sie ein Backup ElastiCache

Jetzt haben Sie Ihren S3-Bucket erstellt und ElastiCache Zugriffsberechtigungen erteilt. Als Nächstes können Sie die ElastiCache Konsole, die oder die verwenden AWS CLI, ElastiCache API um Ihren Snapshot dorthin zu exportieren. In den folgenden Beispielen wird davon ausgegangen, dass die IAM Identität des Aufrufers über die folgenden zusätzlichen S3-spezifischen IAM Berechtigungen verfügt.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListAllMyBuckets",
      "s3:PutObject",
      "s3:GetObject",
      "s3:DeleteObject",
      "s3:ListBucket"
    ],
    "Resource": "arn:aws:s3:::*"
  }]
}
```

Nachfolgend finden Sie ein Beispiel dafür, wie die aktualisierte Richtlinie für das S3-Bucket für Opt-in-Regionen aussehen könnte. (In diesem Beispiel wird die Region Asien-Pazifik (Hongkong) verwendet.)

```
{
  "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/*"
      ]
    }
  ]
}
```

Ein ElastiCache Backup exportieren (Konsole)

Die folgenden Schritte verwenden die ElastiCache Konsole, um ein Backup in einen Amazon S3 S3-Bucket zu exportieren, sodass Sie von außerhalb darauf zugreifen können ElastiCache. Der Amazon S3 S3-Bucket muss sich in derselben AWS Region wie das ElastiCache Backup befinden.

Um ein ElastiCache Backup in einen Amazon S3 S3-Bucket zu exportieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste Ihrer Sicherungen anzuzeigen, wählen Sie im linken Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, die Sie exportieren möchten.
4. Wählen Sie die Option Kopieren aus.
5. Gehen Sie im Dialogfeld Create a Copy of the Backup (Eine Kopie der Sicherung erstellen?) wie folgt vor:
 - a. Geben Sie in das Feld New backup name einen Namen für die neue Sicherung ein.

Der Name muss zwischen 1 und 1.000 Zeichen lang sein und kann UTF -8 kodiert werden.

ElastiCache fügt dem hier eingegebenen Wert eine Instanz-ID und `.rdb` hinzu. Wenn Sie z. B. `my-exported-backup` eingeben, erstellt ElastiCache `my-exported-backup-0001.rdb`.

- b. Wählen Sie aus der Liste S3-Zielspeicherort den Namen des Amazon-S3-Buckets aus, in den Sie Ihre Sicherung kopieren möchten (der von Ihnen unter [Erstellen eines Amazon-S3-Buckets](#) erstellte Bucket).

Der Ziel-S3-Standort muss ein Amazon S3 S3-Bucket in der AWS Region des Backups mit den folgenden Berechtigungen sein, damit der Exportvorgang erfolgreich ist.

- Objektzugriff – Lesen und Schreiben.
- Berechtigungszugriff – Lesen.

Weitere Informationen finden Sie unter [Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket](#).

- c. Wählen Sie die Option Kopieren aus.

Note

Wenn Ihr S3-Bucket nicht über die erforderlichen Berechtigungen verfügt, ElastiCache um ein Backup dorthin zu exportieren, erhalten Sie eine der folgenden Fehlermeldungen. Kehren Sie zu [Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket](#) zurück, um die angegebenen Berechtigungen hinzuzufügen, und exportieren Sie Ihre Sicherung erneut.

- ElastiCache hat keine READ Berechtigungen %s für den S3-Bucket erhalten.

Lösung: Fügen Sie Leseberechtigungen für den Bucket hinzu.

- ElastiCache wurden keine WRITE Berechtigungen %s für den S3-Bucket erteilt.

Lösung: Fügen Sie Schreibberechtigungen für den Bucket hinzu.

- ElastiCache wurden keine READ _ ACP Berechtigungen %s für den S3-Bucket erteilt.

Lösung: Fügen Sie Read-Zugriff für Berechtigungen für den Bucket hinzu.

Wenn Sie Ihr Backup in eine andere AWS Region kopieren möchten, verwenden Sie Amazon S3, um es zu kopieren. Weitere Informationen finden Sie unter [Kopieren von Objekten](#) im Entwicklerhandbuch zu Amazon Simple Storage Service.

Exportieren eines ElastiCache serverlosen Backups ()AWS CLI

Exportieren des Backups eines Serverless-Caches

Exportieren Sie das Backup mithilfe des `export-serverless-cache-snapshot` CLI Vorgangs mit den folgenden Parametern in einen Amazon S3 S3-Bucket:

Parameter

- `--serverless-cache-snapshot-name` Name der zu kopierenden Sicherung.
- `--s3-bucket-name` – Name des Amazon-S3-Buckets, zu dem die Sicherung exportiert werden soll. Im angegebenen Bucket wird eine Kopie der Sicherung erstellt.

Damit der Exportvorgang erfolgreich ist, `--s3-bucket-name` muss es sich um einen Amazon S3 S3-Bucket in der AWS Region des Backups mit den folgenden Berechtigungen handeln.

- Objektzugriff – Lesen und Schreiben.
- Berechtigungszugriff – Lesen.

Die folgende Operation kopiert eine Sicherung zu `my-s3-bucket`.

Für Linux, macOS oder Unix:

```
aws elasticache export-serverless-cache-snapshot \  
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 \  
  --s3-bucket-name my-s3-bucket
```

Für Windows:

```
aws elasticache export-serverless-cache-snapshot ^  
  --serverless-cache-snapshot-name automatic.my-redis-2023-11-27 ^  
  --s3-bucket-name my-s3-bucket
```

Exportieren eines selbst entworfenen ElastiCache Cluster-Backups ()AWS CLI

Exportieren eines Backups eines selbst entworfenen Clusters

Exportieren Sie das Backup mithilfe des `copy-snapshot` CLI Vorgangs mit den folgenden Parametern in einen Amazon S3 S3-Bucket:

Parameter

- `--source-snapshot-name` Name der zu kopierenden Sicherung.
- `--target-snapshot-name` Name der Sicherungskopie.

Der Name muss zwischen 1 und 1.000 Zeichen lang sein und kann UTF -8 kodiert werden.

ElastiCache fügt dem hier eingegebenen Wert eine Instanz-ID und `.rdb` hinzu. Wenn Sie z. B. `my-exported-backup` eingeben, erstellt ElastiCache `my-exported-backup-0001.rdb`.

- `--target-bucket` – Name des Amazon-S3-Buckets, zu dem die Sicherung exportiert werden soll. Im angegebenen Bucket wird eine Kopie der Sicherung erstellt.

Damit der Exportvorgang erfolgreich ist, `--target-bucket` muss es sich um einen Amazon S3 S3-Bucket in der AWS Region des Backups mit den folgenden Berechtigungen handeln.

- Objektzugriff – Lesen und Schreiben.
- Berechtigungszugriff – Lesen.

Weitere Informationen finden Sie unter [Gewähren Sie ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket](#).

Die folgende Operation kopiert eine Sicherung zu `my-s3-bucket`.

Für Linux, macOS oder 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
```

Für 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
```

Wiederherstellen aus einem Backup in einen neuen Cache

Sie können ein vorhandenes Backup von Valkey in einem neuen Valkey-Cache oder einem selbst entworfenen Cluster wiederherstellen und ein vorhandenes OSS Redis-Backup in einem neuen OSS Redis-Cache oder einem selbst entworfenen Cluster wiederherstellen. Sie können ein vorhandenes Memcached-Cache-Backup auch in einem neuen serverlosen Memcached-Cache wiederherstellen.

Wiederherstellen eines Backups in einem Serverless-Cache (Konsole)

Note

ElastiCache Serverless unterstützt RDB Dateien, die mit Valkey 7.2 und höher kompatibel sind, sowie OSS Redis-Versionen zwischen 5.0 und der neuesten verfügbaren Version.

So stellen Sie ein Backup in einem Serverless-Cache wieder her (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Backups das Kontrollkästchen links neben dem Namen des Backups, das Sie wiederherstellen möchten.
4. Wählen Sie Aktionen und anschließend Wiederherstellen aus.
5. Geben Sie einen Namen und eine optionale Beschreibung für den neuen Serverless-Cache ein.
6. Klicken Sie auf Erstellen, um Ihren neuen Cache zu erstellen und Daten aus Ihrem Backup zu importieren.

Wiederherstellen eines Backups in einem selbst entworfenen Cluster (Konsole)

So stellen Sie ein Backup in einem selbst entworfenen Cluster wieder her (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Backups aus.
3. Aktivieren Sie in der Liste der Sicherungen das Kontrollkästchen links neben dem Namen der Sicherung, aus der Sie wiederherstellen möchten.

4. Wählen Sie Aktionen und anschließend Wiederherstellen aus.
5. Wählen Sie Entwerfen Sie Ihren eigenen Cache aus und passen Sie die Cluster-Einstellungen an, z. B. Knotentyp, Größe, Anzahl der Shards, Replikate, AZ-Platzierung und Sicherheitseinstellungen.
6. Klicken Sie auf Erstellen, um Ihren neuen selbst entworfenen Cache zu erstellen und Daten aus Ihrem Backup zu importieren.

Wiederherstellen eines Backups in einem Serverless-Cache (AWS CLI)

Note

ElastiCache Serverless unterstützt RDB Dateien, die mit Valkey 7.2 und höher kompatibel sind, sowie OSS Redis-Versionen zwischen 5.0 und der neuesten verfügbaren Version.

So stellen Sie ein Backup in einem neuen Serverless-Cache wieder her (AWS CLI)

Im folgenden AWS CLI Beispiel wird ein neuer Cache erstellt, der Daten aus einem `create-serverless-cache` Backup verwendet und daraus importiert.

Für Linux, macOS oder Unix:

```
aws elasticache create-serverless-cache \  
  
  --serverless-cache-name CacheName \  
  --engine redis  
  --snapshot-arns-to-restore Snapshot-ARN
```

Für Windows:

```
aws elasticache create-serverless-cache ^  
  
  --serverless-cache-name CacheName ^  
  --engine redis ^  
  --snapshot-arns-to-restore Snapshot-ARN
```

Wiederherstellen eines Backups in einem selbst entworfenen Cluster (AWS CLI)

So stellen Sie ein Backup in einem selbst entworfenen Cluster wieder her (AWS CLI)

Sie können ein Valkey- oder Redis-Backup OSS (Clustermodus deaktiviert) auf zwei Arten wiederherstellen.

- ```
aws elasticache create-serverless-cache \
 --serverless-cache-name CacheName \
 --engine redis \
 --snapshot-arns-to-restore Snapshot-ARN
```

- Für Windows:

```
aws elasticache create-serverless-cache ^ \
 --serverless-cache-name CacheName ^ \
 --engine redis ^ \
 --snapshot-arns-to-restore Snapshot-ARN
```

Wiederherstellen eines Backups in einem selbst entworfenen Cluster (AWS CLI)

So stellen Sie ein Backup in einem selbst entworfenen Cluster wieder her (AWS CLI)

Sie können ein OSS serverloses Valkey- oder Redis-Cache-Backup wiederherstellen, und Sie können auch einen selbst entworfenen Valkey- oder Redis-Cluster wiederherstellen. OSS

Sie können ein serverloses Valkey- oder OSS Redis-Cache-Backup auf zwei Arten wiederherstellen.

- Mithilfe dieses Vorgangs können Sie eine Wiederherstellung auf einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem Knoten durchführen. AWS CLI `create-cache-cluster`
- Sie können die Wiederherstellung auf einem Valkey- oder OSS Redis-Cluster mit Read Replicas (einer Replikationsgruppe) durchführen. Zu diesem Zweck können Sie für den Vorgang entweder Valkey oder Redis OSS (Clustermodus deaktiviert) oder Valkey oder Redis OSS (Clustermodus aktiviert) verwenden. AWS CLI `create-replication-group` In diesem Fall starten Sie die Wiederherstellung mit einer Valkey- oder Redis-.rdb-Datei. OSS Weitere Informationen zum Seeding eines neuen selbst entworfenen Clusters finden Sie unter [Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup](#).

Sie können ein Valkey- oder Redis-Backup OSS (Clustermodus deaktiviert) auf zwei Arten wiederherstellen.

- Sie können mithilfe dieses Vorgangs eine Wiederherstellung auf einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem Knoten durchführen. AWS CLI `create-cache-cluster`

- Sie können die Wiederherstellung auf einem Valkey- oder OSS Redis-Cluster mit Read Replicas (einer Replikationsgruppe) durchführen. Zu diesem Zweck können Sie für den Vorgang entweder Valkey oder Redis OSS (Clustermodus deaktiviert) oder Valkey oder Redis OSS (Clustermodus aktiviert) verwenden. AWS CLI `create-replication-group` In diesem Fall starten Sie die Wiederherstellung mit einer Valkey- oder Redis-.rdb-Datei. OSS Weitere Informationen zum Seeding eines neuen selbst entworfenen Clusters finden Sie unter [Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup](#).

Achten Sie bei Verwendung der Operation `create-cache-cluster` oder `create-replication-group` darauf, den Parameter `--snapshot-name` oder `--snapshot-arns` einzuschließen, um den neuen Cluster oder die Replikationsgruppe mit den Daten aus dem Backup zu starten.

## Löschen einer Sicherung

Eine automatische Sicherung wird automatisch gelöscht, wenn ihre Aufbewahrungsfrist abläuft. Wenn Sie einen Cluster löschen, werden alle seine automatischen Sicherungen ebenfalls gelöscht. Wenn Sie eine Replikationsgruppe löschen, werden alle automatischen Sicherungen von den Clustern in der betreffenden Gruppe ebenfalls gelöscht.

ElastiCache bietet einen API Löschvorgang, mit dem Sie ein Backup jederzeit löschen können, unabhängig davon, ob das Backup automatisch oder manuell erstellt wurde. Da für manuelle Sicherungen keine Aufbewahrungsfrist gilt, können sie nur durch manuelles Löschen entfernt werden.

Sie können ein Backup mit der ElastiCache Konsole AWS CLI, dem oder dem löschen ElastiCache API.

### Löschen einer Sicherung (Konsole)

Mit dem folgenden Verfahren wird ein Backup mithilfe der ElastiCache Konsole gelöscht.

So löschen Sie ein Backup

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Sicherungen aus.

Der Bildschirm „Backups“ wird mit einer Liste Ihrer Sicherungen angezeigt.

3. Aktivieren Sie das Kontrollkästchen links neben dem Namen der Sicherung, die Sie löschen möchten.

4. Wählen Sie Löschen.
5. Wenn Sie diese Sicherung löschen möchten, wählen Sie auf dem Bestätigungsbildschirm Delete Backup die Option Delete aus. Der Status wird in deleting geändert.

### Löschen eines Serverless-Backups (AWS CLI)

Verwenden Sie den AWS CLI Vorgang delete-snapshot mit dem folgenden Parameter, um ein serverloses Backup zu löschen.

- `--serverless-cache-snapshot-name` Name der zu löschenden Sicherung.

Der folgende Code löscht die Sicherung myBackup.

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-name myBackup
```

Weitere Informationen finden Sie [delete-serverless-cache-snapshot](#) in der Befehlsreferenz.AWS CLI

### Löschen eines selbst entworfenes Cluster-Backups (AWS CLI)

Verwenden Sie den AWS CLI Vorgang delete-snapshot mit dem folgenden Parameter, um ein selbst entworfenes Cluster-Backup zu löschen.

- `--snapshot-name` Name der zu löschenden Sicherung.

Der folgende Code löscht die Sicherung myBackup.

```
aws elasticache delete-snapshot --snapshot-name myBackup
```

Weitere Informationen finden Sie unter [delete-snapshot](#) in derAWS CLI -CLI-Befehlsreferenz.

## Markieren von Sicherungen

Sie können -Tresoren Metadaten in Form von Tags zuweisen. Mit Tags (Markierungen) können Sie Ihre -Ressourcen auf unterschiedliche Weise kategorisieren (z. B. nach Zweck, Eigentümer oder Umgebung). Dies ist nützlich, wenn Sie viele Ressourcen desselben Typs haben — In diesem Fall können Sie schnell bestimmte Ressourcen basierend auf den zugewiesenen Tags (Markierungen) bestimmen. Weitere Informationen finden Sie unter [Ressourcen, die markiert werden können](#).

Mithilfe von Tags zur Kostenzuweisung können Sie Ihre Kosten für mehrere AWS Dienste verfolgen, indem Sie Ihre Ausgaben auf Rechnungen nach Tagwerten gruppieren. Weitere Informationen zu Kostenzuordnungs-Tags finden Sie unter [Verwenden von Kostenzuordnungs-Tags](#).

Mithilfe der ElastiCache Konsole können ElastiCache API Sie Kostenzuordnungs-Tags zu Ihren Backups hinzufügen, auflisten, ändern, entfernen oder kopieren. AWS CLI Weitere Informationen finden Sie unter [Überwachung von Kosten mit Kostenzuordnungs-Tags](#).

## Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup

Wenn Sie einen neuen OSS selbst entworfenen Valkey- oder Redis-Cluster erstellen, können Sie ihn mit Daten aus einer Valkey- oder Redis-.rdb-Backup-Datei versorgen. OSS Das Seeding des Clusters ist nützlich, wenn Sie derzeit eine Valkey- oder OSS Redis-Instanz außerhalb von verwalten und Ihren neuen ElastiCache (Redis) selbst entworfenen Cluster mit Ihren vorhandenen Valkey ElastiCache - oder OSS Redis-Daten füllen möchten. OSS

Informationen zum Seeding eines neuen, OSS selbst entworfenen Valkey- oder Redis-Clusters aus einem in Amazon erstellten Valkey- oder OSS Redis-Backup finden Sie unter [ElastiCache Wiederherstellen aus einem Backup in einen neuen Cache](#)

Wenn Sie eine Valkey- oder OSS Redis-.rdb-Datei für das Seeding eines neuen, selbst entworfenen Clusters verwenden, können Sie Folgendes tun:

- Führen Sie ein Upgrade von einem nicht partitionierten Cluster auf einen selbst entworfenen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) durch, auf dem Redis Version 3.2.4 ausgeführt wird. OSS
- Geben Sie eine Anzahl von Shards (im und als Knotengruppen bezeichnet) im neuen, selbst entworfenen Cluster an API. CLI Diese Zahl kann von der Anzahl an Shards im selbst entworfenen Cluster abweichen, mit der die Backup-Datei erstellt wurde.
- Geben Sie einen anderen Knotentyp für den neuen selbst entworfenen Cluster an, und zwar größer oder kleiner als der, der in dem dem Backup zugrunde liegenden Cluster verwendet wurde. Wenn Sie auf einen kleineren Knotentyp skalieren, stellen Sie sicher, dass der neue Knotentyp über ausreichend Speicher für Ihre Daten und Valkey- oder Redis-Overhead verfügt. OSS Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#).
- Verteilen Sie Ihre Schlüssel in den Steckplätzen des neuen Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) anders als in dem Cluster, der zur Erstellung der Backup-Datei verwendet wurde.

**Note**

Sie können einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) nicht anhand einer .rdb-Datei starten, die aus einem Valkey- oder OSS Redis-Cluster (Clustermodus aktiviert) erstellt wurde.

**⚠ Important**

- Sie müssen sicherstellen, dass Ihre Valkey- oder OSS Redis-Backup-Daten die Ressourcen des Knotens nicht überschreiten. Sie können beispielsweise keine .rdb-Datei mit 5 GB Valkey- oder OSS Redis-Daten auf einen Cache.m3.medium-Knoten mit 2,9 GB Arbeitsspeicher hochladen.

Wenn die Sicherung zu groß ist, lautet der Status des resultierenden Clusters `restore-failed`. In diesem Fall müssen Sie den Cluster löschen und von Neuem beginnen.

Eine vollständige Liste der Knotentypen und Spezifikationen finden Sie unter [OSSRedis-Knotentyp-spezifische Parameter](#) und zu [ElastiCache Amazon-Produktfunktionen und -details](#).

- Sie können eine Valkey- oder OSS Redis-.rdb-Datei nur mit der serverseitigen Amazon S3 S3-Verschlüsselung (-S3) verschlüsseln. SSE Weitere Informationen finden Sie unter [Schützen von Daten mithilfe serverseitiger Verschlüsselung](#).

Im Folgenden finden Sie Themen, die Sie durch die Migration Ihres Clusters von außerhalb ElastiCache für Valkey oder Redis zu (Redis) führen. OSS ElastiCache OSS

**Migration zu (Redis) ElastiCache OSS**

- [Schritt 1: Erstellen Sie ein Valkey- oder Redis-Backup OSS](#)
- [Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners](#)
- [Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3](#)
- [Schritt 4: Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei](#)

**Migration von externen Diensten zu ElastiCache (Redis). OSS**

- [Schritt 1: Erstellen Sie ein Valkey- oder Redis-Backup OSS](#)

- [Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners](#)
- [Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3](#)
- [Schritt 4: Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei](#)

## Schritt 1: Erstellen Sie ein Valkey- oder Redis-Backup OSS

Um das Valkey- oder OSS Redis-Backup zu erstellen, um Ihre ElastiCache (Redis-) Instanz zu starten OSS

1. Connect zu Ihrer vorhandenen Valkey- oder OSS Redis-Instanz her.
2. Führen Sie einen der beiden BGSAVE SAVE Operationen aus, um ein Backup zu erstellen. Achten Sie auf den Speicherort der .rdb-Datei.

BGSAVE ist asynchron und blockiert während der Verarbeitung keine anderen Clients. Weitere Informationen finden Sie [BGSAVE](#) auf der Valkey-Website.

SAVE ist synchron und blockiert andere Vorgänge, bis sie Verarbeitung abgeschlossen ist. Weitere Informationen finden Sie [SAVE](#) auf der Valkey-Website.

Weitere Informationen zum Erstellen eines Backups finden Sie unter [Persistence](#) auf der Valkey-Website.

## Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners

Wenn Sie die Sicherungsdatei erstellt haben, müssen Sie sie zu einem Ordner innerhalb eines Amazon-S3-Buckets hochladen. Hierzu müssen bereits ein Amazon-S3-Bucket und ein Ordner innerhalb dieses Buckets vorhanden sein. Wenn Sie bereits einen Amazon-S3-Bucket und Ordner mit den entsprechenden Berechtigungen besitzen, können Sie mit [Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3](#) fortfahren.

So erstellen Sie einen Amazon-S3-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Führen Sie die Anweisungen zum Erstellen eines Amazon S3 Buckets unter [Erstellen eines Buckets](#) im Benutzerhandbuch zu Amazon Simple Storage Service aus.

Der Name Ihres Amazon S3 S3-Buckets muss DNS -konform sein. Andernfalls ElastiCache kann nicht auf Ihre Backup-Datei zugreifen. Die Regeln für die DNS Einhaltung der Vorschriften lauten wie folgt:

- Bucket-Namen müssen mindestens 3 und dürfen höchstens 63 Zeichen umfassen.
- Die Namen müssen eine Folge aus einer oder mehreren Beschriftungen darstellen, die durch einen Punkt (.) getrennt sind, wobei jede Beschriftung:
  - mit einem Kleinbuchstaben oder einer Zahl beginnen.
  - mit einem Kleinbuchstaben oder einer Zahl beginnen.
  - Enthält nur Kleinbuchstaben, Zahlen und Bindestriche.
- Er darf nicht als IP-Adresse (z. B. 192.0.2.0) formatiert sein.

Sie müssen Ihren Amazon S3 S3-Bucket in derselben AWS Region wie Ihr neuer ElastiCache (Redis-OSS) Cluster erstellen. Dieser Ansatz stellt sicher, dass beim ElastiCache Lesen Ihrer RDB-Datei aus Amazon S3 die höchste Datenübertragungsgeschwindigkeit erreicht wird.

 Note

Um Ihre Daten so sicher wie möglich zu halten, sollten Sie die Berechtigungen für Ihr Amazon-S3-Bucket so restriktiv wie möglich gestalten. Gleichzeitig müssen die Berechtigungen weiterhin zulassen, dass der Bucket und sein Inhalt für das Seeding Ihres neuen Valkey- oder Redis-Clusters verwendet werden können. OSS

So fügen Sie einem Amazon S3 Bucket einen Ordner hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, in den die .rdb-Datei hochgeladen werden soll.
3. Wählen Sie Create folder.
4. Geben Sie einen Namen für den neuen Ordner ein.
5. Wählen Sie Save (Speichern) aus.

Notieren Sie sich sowohl den Bucketnamen als auch den Ordernamen.

## Schritt 3: Hochladen Ihrer Sicherung auf Amazon S3

Laden Sie jetzt die .rdb-Datei hoch, die Sie in [Schritt 1: Erstellen Sie ein Valkey- oder Redis-Backup OSS](#) erstellt haben. Sie laden sie in den Amazon-S3-Bucket und -Ordner hoch, die sie in [Schritt 2: Erstellen eines Amazon-S3-Buckets und -Ordners](#) erstellt haben. Weitere Informationen zu dieser Aufgabe finden Sie unter [Objekte in einen Bucket einfügen](#). Wählen Sie zwischen den Schritten 2 und 3 den Namen des von Ihnen erstellten Ordners aus.

So laden Sie Ihre .rdb-Datei in einen Amazon-S3-Ordner hoch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Amazon-S3-Bucket aus, den Sie in Schritt 2 erstellt haben.
3. Wählen Sie den Namen des Ordners aus, den Sie in Schritt 2 erstellt haben.
4. Klicken Sie auf Hochladen.
5. Klicken Sie auf Add files.
6. Navigieren Sie zu der Datei oder den Dateien, die Sie hochladen möchten, und wählen Sie dann die Datei oder die Dateien aus. Halten Sie zur Auswahl mehrerer Dateien die Strg-Taste während der Auswahl der Dateinamen gedrückt.
7. Klicken Sie auf Open.
8. Bestätigen Sie, dass die korrekte Datei bzw. die korrekten Dateien im Dialogfeld Upload aufgelistet werden. Wählen Sie dann Upload.

Notieren Sie den Pfad zu Ihrer .rdb-Datei. Wenn der Bucket-Name z. B. myBucket und der Pfad myFolder/redis.rdb lautet, geben Sie myBucket/myFolder/redis.rdb ein. Sie benötigen diesen Pfad zum Starten des neuen Clusters mit den Daten in dieser Sicherung.

Weitere Informationen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Entwicklerhandbuch zu Amazon Simple Storage Service.

## Schritt 4: Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei

Gewähren ElastiCache Sie jetzt Lesezugriff auf Ihre .rdb-Backup-Datei. Sie gewähren ElastiCache Zugriff auf Ihre Backup-Datei auf unterschiedliche Weise, je nachdem, ob sich Ihr Bucket in einer AWS Standardregion oder einer AWS Opt-in-Region befindet.

AWS Regionen, die vor dem 20. März 2019 eingeführt wurden, sind standardmäßig aktiviert. Sie können sofort mit der Arbeit in diesen AWS Regionen beginnen. Regionen, die nach dem 20. März 2019 eingeführt werden, wie Asien-Pazifik (Hongkong) und Naher Osten (Bahrain), sind standardmäßig deaktiviert. Sie müssen diese Regionen aktivieren oder sich dafür anmelden, bevor Sie sie verwenden können, wie unter [Verwalten von AWS -Regionen](#) im Allgemeine AWS-Referenz beschrieben.

Wählen Sie Ihren Ansatz je nach AWS Region:

- Verwenden Sie für eine Standardregion das Verfahren unter [Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Standardregion](#).
- Verwenden Sie für eine Opt-In-Region das Verfahren unter [Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Opt-in-Region](#).

Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Standardregion

AWS Regionen, die vor dem 20. März 2019 eingeführt wurden, sind standardmäßig aktiviert. Sie können sofort mit der Arbeit in diesen AWS Regionen beginnen. Regionen, die nach dem 20. März 2019 eingeführt werden, wie Asien-Pazifik (Hongkong) und Naher Osten (Bahrain), sind standardmäßig deaktiviert. Sie müssen diese Regionen aktivieren oder sich dafür anmelden, bevor Sie sie verwenden können, wie unter [Verwalten von AWS -Regionen](#) im Allgemeine AWS-Referenz beschrieben.

Um ElastiCache Lesezugriff auf die Backup-Datei in einer AWS Region zu gewähren, die standardmäßig aktiviert ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des S3-Buckets aus, in dem sich Ihre .rdb-Datei befindet.
3. Wählen Sie den Namen des Ordners aus, in dem sich Ihre .rdb-Datei befindet.
4. Wählen Sie den Namen Ihrer .rdb-Sicherungsdatei aus. Der Name der ausgewählten Datei erscheint oberhalb der Registerkarten oben auf der Seite.
5. Wählen Sie Permissions (Berechtigungen).
6. Wenn aws-scs-s3-readonly oder einer der kanonischen Benutzer IDs in der folgenden Liste nicht als Benutzer aufgeführt ist, gehen Sie wie folgt vor:
  - a. Wählen Sie unter Zugriff für andere AWS Konten die Option Empfänger hinzufügen aus.

b. Fügen Sie in dem Feld die kanonische ID der AWS Region wie folgt hinzu:

- AWS GovCloud Region (USA West):

```
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
```

 **Important**

Das Backup muss sich in einem S3-Bucket befinden, damit Sie es auf einen Valkey- oder OSS Redis-Cluster herunterladen können. AWS GovCloud (US)  
AWS GovCloud (US)

- AWS Standardmäßig aktivierte Regionen:

```
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353
```

c. Richten Sie die Berechtigungen für den Bucket durch Auswahl von Yes (Ja) für die folgenden Optionen ein:

- List/write object (Objekt auflisten/schreiben)
- Objektberechtigungen zum Lesen/Schreiben ACL

d. Wählen Sie Save (Speichern) aus.

7. Wählen Sie Overview (Übersicht) und dann Download (Herunterladen).

Gewähren ElastiCache Sie Lesezugriff auf die RDB-Datei in einer Opt-in-Region

AWS Regionen, die vor dem 20. März 2019 eingeführt wurden, sind standardmäßig aktiviert. Sie können sofort mit der Arbeit in diesen AWS Regionen beginnen. Regionen, die nach dem 20. März 2019 eingeführt werden, wie Asien-Pazifik (Hongkong) und Naher Osten (Bahrain), sind standardmäßig deaktiviert. Sie müssen diese Regionen aktivieren oder sich dafür anmelden, bevor Sie sie verwenden können, wie unter [Verwalten von AWS -Regionen](#) im Allgemeine AWS-Referenz beschrieben.

Gewähren Sie jetzt ElastiCache Lesezugriff auf Ihre .rdb-Backup-Datei.

## Um ElastiCache Lesezugriff auf die Sicherungsdatei zu gewähren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des S3-Buckets aus, in dem sich Ihre .rdb-Datei befindet.
3. Wählen Sie den Namen des Ordners aus, in dem sich Ihre .rdb-Datei befindet.
4. Wählen Sie den Namen Ihrer .rdb-Sicherungsdatei aus. Der Name der ausgewählten Datei erscheint oberhalb der Registerkarten oben auf der Seite.
5. Wählen Sie die Registerkarte Berechtigungen.
6. Wählen Sie unter Permissions (Berechtigungen) Bucket policy (Bucket-Richtlinie) aus und klicken Sie dann auf Edit (Bearbeiten).
7. Aktualisieren Sie die Richtlinie, um die ElastiCache erforderlichen Berechtigungen zur Durchführung von Vorgängen zu gewähren:
  - Fügen Sie [ "Service" : "*region-full-name*.elasticache-snapshot.amazonaws.com" ] zu Principal hinzu.
  - Fügen Sie die folgenden Berechtigungen hinzu, die für das Exportieren eines Snapshots in den Amazon-S3-Bucket erforderlich sind:
    - "s3:GetObject"
    - "s3:ListBucket"
    - "s3:GetBucketAcl"

Nachfolgend finden Sie ein Beispiel dafür, wie die aktualisierte Richtlinie aussehen könnte.

```
{
 "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. Wählen Sie Änderungen speichern.

Den ElastiCache Cluster mit den Daten der .rdb-Datei versorgen

Jetzt sind Sie bereit, einen ElastiCache Cluster zu erstellen und ihn mit den Daten aus der RDB-Datei zu versorgen. Folgen Sie zum Erstellen des Clusters den Anweisungen unter [Einen Cluster für Valkey oder Redis erstellen OSS](#) oder [Eine Valkey- oder OSS Redis-Replikationsgruppe von Grund auf neu erstellen](#). Achten Sie darauf, Valkey oder Redis OSS als Cluster-Engine zu wählen.

Die Methode, mit der Sie angeben, ElastiCache wo sich das Backup befindet, das Sie auf Amazon S3 hochgeladen haben, hängt von der Methode ab, mit der Sie den Cluster erstellt haben:

Füllen Sie den ElastiCache (Redis-OSS) Cluster oder die Replikationsgruppe mit den Daten der .rdb-Datei

- Verwenden der Konsole ElastiCache

Wählen Sie bei der Auswahl der Cluster settings (Cluster-Einstellungen) die Option Restore from backups (Aus Backups wiederherstellen) als Methode zur Cluster-Erstellung aus, und wählen Sie dann im Abschnitt Backup source (Backup-Quelle) die Option Other backups (Andere Backups) als Source (Quelle) aus. Geben Sie im Feld S3-Speicherort der RDB Seed-Datei den Amazon S3 S3-Pfad für die Dateien ein. Wenn Sie mehrere .rdb-Dateien besitzen, geben Sie den Pfad für jede Datei in eine durch Kommas getrennten Liste ein. Der Amazon-S3-Pfad sieht etwa aus wie *myBucket/myFolder/myBackupFilename.rdb*.

- Verwenden Sie den AWS CLI

Wenn Sie die `create-replication-group` Operation `create-cache-cluster` oder verwenden, verwenden Sie den Parameter, `--snapshot-arns` um ARN für jede RDB-Datei eine vollständig qualifizierte Datei anzugeben. Beispiel,

`arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb`. Sie ARN müssen auf die Backup-Dateien zugreifen, die Sie in Amazon S3 gespeichert haben.

- Mit dem ElastiCache API

Wenn Sie die `CreateReplicationGroup` ElastiCache API Operation `CreateCacheCluster` oder verwenden, verwenden Sie den Parameter, `SnapshotArns` um ARN für jede RDB-Datei eine vollständig qualifizierte Datei anzugeben. Beispiel, `arn:aws:s3:::myBucket/myFolder/myBackupFilename.rdb`. Sie ARN müssen auf die Backup-Dateien zugreifen, die Sie in Amazon S3 gespeichert haben.

### Important

Beim Seeding eines Valkey- oder Redis-Clusters OSS (Cluster-Modus aktiviert) müssen Sie jede Knotengruppe (Shard) im neuen Cluster oder in der neuen Replikationsgruppe konfigurieren. Verwenden Sie dazu den Parameter `--node-group-configuration` (API:`NodeGroupConfiguration`). Weitere Informationen finden Sie hier:

- CLI: [create-replication-group](#) in der AWS CLI Referenz
- API: [CreateReplicationGroup](#) in der ElastiCache API Referenz

Während der Erstellung Ihres Clusters werden die Daten in Ihrem Valkey- oder OSS Redis-Backup in den Cluster geschrieben. Sie können den Fortschritt überwachen, indem Sie sich die ElastiCache Ereignismeldungen ansehen. Rufen Sie dazu die ElastiCache Konsole auf und wählen Sie Cache Events aus. Sie können auch die AWS ElastiCache Befehlszeilenschnittstelle verwenden oder ElastiCache API um Ereignismeldungen abzurufen. Weitere Informationen finden Sie unter [ElastiCache Ereignisse anzeigen](#).

# Motorversionen und Aufrüstung in ElastiCache

In diesem Abschnitt werden die unterstützten Valkey-, Redis OSS - und Memcached-Engines sowie Informationen zum Upgrade behandelt. Beachten Sie, dass alle mit Redis OSS 7.2 verfügbaren Funktionen standardmäßig in Valkey 7.2 und höher verfügbar sind. Sie können auch von einigen vorhandenen Engines ElastiCache mit Redis auf eine OSS Valkey-Engine aufrüsten.

## Themen

- [Versionsverwaltung für ElastiCache](#)
- [So führen Sie ein Upgrade von Engine-Versionen aus](#)
- [So lösen Sie ein Cross-Engine-Upgrade von Redis auf Valkey OSS aus](#)
- [Unterstützte Engines und Versionen](#)
- [Wesentliche Unterschiede im Versionsverhalten und in der Kompatibilität mit Valkey](#)
- [Wesentliche Unterschiede im Versionsverhalten und in der Kompatibilität mit Redis OSS](#)
- [Blockierte Valkey- oder OSS Redis-Engine-Upgrades beheben](#)

## Versionsverwaltung für ElastiCache

Sie können festlegen, wie Sie Ihre ElastiCache Caches und selbst entworfenen Cluster, die für die Engines Valkey, Redis und Memcached aktualisiert wurden, aktualisieren möchten. OSS

### ElastiCache Versionsverwaltung für Serverless Cache

Verwalten Sie, ob und wann der ElastiCache Serverless Cache aktualisiert wird, und führen Sie Versionsupgrades zu Ihren eigenen Bedingungen und Zeitplänen durch.

ElastiCache Serverless wendet automatisch die neueste Version MINOR und PATCH Softwareversion auf Ihren Cache an, ohne dass dies Auswirkungen oder Ausfallzeiten auf Ihre Anwendung hat. Von Ihrer Seite aus ist keine Aktion erforderlich.

Wenn eine neue MAJOR Version verfügbar ist, sendet Ihnen ElastiCache Serverless eine Benachrichtigung in der Konsole und ein Ereignis in. EventBridge Sie können wählen, ob Sie Ihren Cache auf die neueste Hauptversion aktualisieren möchten, indem Sie Ihren Cache mithilfe der Konsole ändernCLI, oder API und die neueste Engine-Version auswählen.

## Versionsverwaltung für selbst entworfene Cluster ElastiCache

Wenn Sie mit selbst entworfenen ElastiCache Clustern arbeiten, können Sie steuern, wann die Software, die Ihren Cache-Cluster unterstützt, auf neue Versionen aktualisiert wird, die von unterstützt werden. ElastiCache Sie können steuern, wann Ihr Cache auf die neuesten verfügbaren MAJOR Versionen und PATCH aktualisiert werden soll. MINOR Sie starten Engine-Versionen-Upgrades für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie ihn bzw. sie ändern und eine neue Engine-Version angeben.

Sie können steuern, ob und wann die protokollkonforme Software, die Ihren Cache-Cluster unterstützt, auf neue Versionen aktualisiert wird, die von unterstützt werden. ElastiCache Mit diesem Maß an Kontrolle können Sie die Kompatibilität mit bestimmten Versionen aufrechterhalten, neue Versionen mit Ihrer Anwendung testen, bevor Sie sie für die Produktion bereitstellen, und Versions-Upgrades nach Ihren eigenen Vorgaben und Zeitplänen durchführen lassen.

Da Versions-Upgrades ggf. mit Kompatibilitätsrisiken verbunden sind, können sie nicht automatisch eingestellt werden. Sie müssen von Ihnen selbst installiert werden.

### Valkey- und Redis-Cluster OSS

#### Note

- Wenn ein Valkey- oder OSS Redis-Cluster in einer oder mehreren Regionen repliziert wird, wird die Engine-Version für sekundäre Regionen und dann für die primäre Region aktualisiert.
- ElastiCache (Redis-OSS) Versionen werden mit einer semantischen Version identifiziert, die eine UND-Komponente umfasst. MAJOR MINOR In Redis OSS 6.2 ist die Hauptversion beispielsweise 6 und die Nebenversion 2. Beim Betrieb selbst entworfener Cluster macht ElastiCache (RedisOSS) auch die PATCH Komponente verfügbar, z. B. Redis OSS 6.2.1, und die Patch-Version ist 1.

MAJORVersionen sind für API inkompatible Änderungen und MINOR Versionen für neue Funktionen, die abwärtskompatibel hinzugefügt wurden. PATCHVersionen sind für abwärtskompatible Bugfixes und nicht funktionale Änderungen vorgesehen.

Mit Valkey und Redis OSS initiieren Sie Engine-Versionen-Upgrades für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie ihn ändern und eine neue Engine-Version angeben. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

## Memcached

Bei Memcached müssen Sie für ein Upgrade auf eine neuere Version Ihren Cache-Cluster ändern und die neue Engine-Version angeben, die Sie verwenden möchten. Das Upgrade auf eine neuere Memcached-Version ist ein destruktiver Prozess – Sie verlieren Ihre Daten und beginnen mit einem kalten Cache. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

Beim Upgraden einer älteren Memcached-Version auf Memcached Version 1.4.33 oder höher sind folgende Anforderungen zu beachten. `CreateCacheCluster` und `ModifyCacheCluster` schlagen unter den folgenden Bedingungen fehl:

- Wenn `slab_chunk_max > max_item_size`.
- Wenn `max_item_size modulo slab_chunk_max != 0`.
- Wenn `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

Der Wert `(max_cache_memory - memcached_connections_overhead)` ist der für Daten nutzbare Speicher des Knotens. Weitere Informationen finden Sie unter [Overhead von Memcached-Verbindungen](#).

## Überlegungen zum Upgrade bei der Arbeit mit selbst entworfenen Clustern

### Note

Die folgenden Überlegungen gelten nur für Upgrades von selbst entworfenen Clustern. Sie gelten nicht für Serverless. ElastiCache

## Überlegungen zu Valkey und Redis OSS

Beachten Sie beim Upgrade eines selbst entworfenen Valkey- oder OSS Redis-Clusters Folgendes.

- Versionsverwaltung der Engine ist so entwickelt, dass Sie so viel Kontrolle wie möglich darüber haben, wie Patches erfolgt. ElastiCache behält sich jedoch das Recht vor, Ihren Cluster in Ihrem

Namen zu patchen, falls im unwahrscheinlichen Fall eine kritische Sicherheitslücke im System oder in der Cache-Software auftritt.

- Ab Valkey 7.2 und Redis OSS 6.0 ElastiCache wird für jede Nebenversion eine einzige Version angeboten, anstatt mehrere Patch-Versionen anzubieten.
- Ab der OSS Redis-Engine-Version 5.0.6 können Sie Ihre Cluster-Version mit minimaler Ausfallzeit aktualisieren. Der Cluster kann während des gesamten Upgrades gelesen und in der Regel auch beschrieben werden, ausgenommen während der Failover-Operation, der nur einige Sekunden dauert.
- Sie können Ihre ElastiCache Cluster auch mit Versionen vor 5.0.6 aktualisieren. Der Prozess ist derselbe, kann jedoch während der DNS Ausbreitung eine längere Failover-Zeit in Anspruch nehmen (30—1 m).
- Ab Redis OSS 7 wird ElastiCache das Umschalten zwischen Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis (Clustermodus aktiviert) unterstützt. OSS
- Der Upgrade-Prozess der Amazon ElastiCache (RedisOSS) Engine ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich beizubehalten, und erfordert eine erfolgreiche OSS Redis-Replikation.
- Beim Upgrade der Engine ElastiCache werden bestehende Client-Verbindungen beendet. [Um Ausfallzeiten bei Engine-Upgrades zu minimieren, empfehlen wir Ihnen, bewährte Methoden für OSS Redis-Clients mit wiederholten Fehlern und exponentiellem Backoff sowie bewährte Methoden zur Minimierung von Ausfallzeiten während der Wartung zu implementieren.](#)
- Sie können nicht direkt von Valkey oder Redis OSS (Clustermodus deaktiviert) auf Valkey oder Redis (Clustermodus aktiviert) aktualisieren, wenn Sie Ihre Engine OSS aktualisieren. Das folgende Verfahren zeigt Ihnen, wie Sie ein Upgrade von Valkey oder Redis OSS (Clustermodus deaktiviert) auf Valkey oder Redis (Clustermodus aktiviert) durchführen. OSS

So führen Sie ein Upgrade von einer Valkey- oder Redis-Engine-Version OSS (Cluster-Modus deaktiviert) auf eine Valkey- oder OSS Redis-Engine-Version (Clustermodus aktiviert) durch

1. Erstellen Sie eine Sicherungskopie Ihres Valkey- oder Redis-Clusters oder Ihrer Replikationsgruppe OSS (Clustermodus deaktiviert). Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).
2. Verwenden Sie das Backup, um einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mit einem Shard (Knotengruppe) zu erstellen und zu starten. Geben Sie die neue Engine-Version an und aktivieren Sie den Cluster-Modus, wenn Sie den Cluster oder die Replikationsgruppe erstellen. Weitere Informationen finden Sie unter [Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup](#).

3. Löschen Sie den alten Valkey- oder Redis-Cluster oder die Redis-Cluster OSS (Cluster-Modus deaktiviert) oder die alte Replikationsgruppe. Weitere Informationen finden Sie unter [Löschen eines Clusters in ElastiCache](#) oder [Löschen einer Replikationsgruppe](#).
  4. Skalieren Sie den neuen Valkey- oder Redis-Cluster oder die Redis-Cluster OSS (Cluster-Modus aktiviert) oder die neue Replikationsgruppe auf die Anzahl der Shards (Knotengruppen), die Sie benötigen. Weitere Informationen finden Sie unter [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
- Beim Upgrade von Hauptversionen der Engine, beispielsweise von 5.0.6 auf 6.0, müssen Sie auch eine neue Parametergruppe auswählen, die mit der neuen Engine-Version kompatibel ist.
  - Für einzelne OSS Redis-Cluster und Cluster mit deaktiviertem Multi-AZ empfehlen wir, Redis ausreichend Speicher zur Verfügung zu stellen, wie unter beschrieben. OSS [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#) In diesen Fällen steht der primäre Knoten während des Upgrade-Prozesses für Serviceanfragen nicht zur Verfügung.
  - Für OSS Redis-Cluster mit aktiviertem Multi-AZ empfehlen wir außerdem, Engine-Upgrades in Zeiten mit geringem eingehendem Schreibverkehr zu planen. Bei einem Upgrade auf Redis OSS 5.0.6 oder höher steht der primäre Cluster während des Upgrade-Vorgangs weiterhin für Serviceanfragen zur Verfügung.

Cluster und Replikationsgruppen mit mehreren Shards werden wie folgt verarbeitet und gepatcht:

- Alle Shards werden parallel verarbeitet. Es wird jeweils nur eine Upgrade-Operation für einen Shard gleichzeitig durchgeführt.
- In jedem Shard werden alle Replicas verarbeitet, bevor der Primärknoten verarbeitet wird. Wenn es in einem Shard weniger Replicas gibt, kann der Primärknoten in diesem Shard verarbeitet werden, bevor die Verarbeitung der Replicas in anderen Shards abgeschlossen wird.
- Die Primärknoten für alle Shards werden seriell verarbeitet. Es erfolgt jeweils nur ein Upgrade für einen Primärknoten gleichzeitig.
- Wenn die Verschlüsselung in Ihrem aktuellen Cluster oder Ihrer Replikationsgruppe aktiviert ist, können Sie nicht auf eine Engine-Version aktualisieren, die keine Verschlüsselung unterstützt, z. B. von 3.2.6 auf 3.2.10.

## Überlegungen zu Memcached

Beachten Sie beim Upgrade eines selbst entworfenen Memcached-Clusters Folgendes.

- Versionsverwaltung der Engine ist so entwickelt, dass Sie so viel Kontrolle wie möglich darüber haben, wie Patches erfolgt. Behält sich jedoch ElastiCache das Recht vor, Ihren Cluster in Ihrem Namen zu patchen, sollte der unwahrscheinliche Fall eintreten, dass das System oder die Cache-Software eine kritische Sicherheitslücke aufweist.
- Da die Memcached-Engine keine Persistenz unterstützt, stellen Versions-Upgrades der Memcached-Engine immer einen Störfall dar, bei dem alle Cache-Daten im Cluster gelöscht werden.

## So führen Sie ein Upgrade von Engine-Versionen aus

### Valkey und Redis OSS

Mit Valkey und Redis OSS initiieren Sie Versionsupgrades für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie ihn mithilfe der ElastiCache Konsole, der oder der AWS CLI ändern ElastiCache API und eine neuere Engine-Version angeben. Weitere Informationen finden Sie unter den folgenden Themen.

| So ändern Sie Cluster und Replikationsgruppen                           |                                                             |
|-------------------------------------------------------------------------|-------------------------------------------------------------|
| Cluster                                                                 | Replikationsgruppen                                         |
| <a href="#">Unter Verwendung der ElastiCache AWS Management Console</a> | <a href="#">Unter Verwendung der AWS Management Console</a> |
| <a href="#">Verwenden Sie das mit AWS CLI ElastiCache</a>               | <a href="#">Verwenden des AWS CLI</a>                       |
| <a href="#">Mit dem ElastiCache API</a>                                 | <a href="#">Verwenden des ElastiCache API</a>               |

### Memcached

Um mit Memcached Versionsupgrades für Ihren Cluster zu starten, ändern Sie ihn und geben eine neuere Engine-Version an. Sie können dies mit der ElastiCache Konsole, dem oder dem AWS CLI tun: ElastiCache API

- Informationen zur Verwendung von AWS Management Console finden Sie unter —[Unter Verwendung der ElastiCache AWS Management Console](#).
- Informationen zur Verwendung von AWS CLI finden Sie unter [Verwenden Sie das mit AWS CLI ElastiCache](#).

- Informationen zur Verwendung von ElastiCache API finden Sie unter [Mit dem ElastiCache API](#).

## So lösen Sie ein Cross-Engine-Upgrade von Redis auf Valkey OSS aus

Sie können eine bestehende OSS Redis-Replikationsgruppe (v4 oder höher) mithilfe der Konsole auf die Valkey-Engine aktualisieren, oder. API CLI

### Note

Wenn Sie einen vorhandenen Redis-Cluster OSS (Clustermodus deaktiviert) mit einem Knoten auf die Valkey-Engine aktualisieren möchten, müssen Sie ihn zunächst einer Replikationsgruppe hinzufügen, indem Sie die folgenden erforderlichen Schritte ausführen: [Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters](#)

Wenn Sie über eine bestehende OSS Redis-Replikationsgruppe verfügen, die die Standard-Cache-Parametergruppe verwendet, können Sie ein Upgrade auf Valkey durchführen, indem Sie die neue Engine und Engine-Version mit angeben. `modify-replication-group` API

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --engine valkey \
 --engine-version 7.2
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --engine valkey ^
 --engine-version 7.2
```

Wenn Sie eine benutzerdefinierte Cache-Parametergruppe auf die bestehende Redis-Replikationsgruppe angewendet haben, die Sie aktualisieren möchten, müssen Sie in der Anfrage auch eine benutzerdefinierte Valkey-Cache-Parametergruppe übergeben. Die benutzerdefinierte Valkey-Eingabeparametergruppe muss dieselben statischen Redis-Parameterwerte haben wie die vorhandene benutzerdefinierte Redis-Parametergruppe.

## Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --engine valkey \
 --engine-version 7.2 \
 --cache-parameter-group-name myParamGroup
```

## Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --engine valkey ^
 --engine-version 7.2 ^
 --cache-parameter-group-name myParamGroup
```

## Engine-übergreifendes Upgrade von Redis OSS auf Valkey für Serverless ElastiCache

Sie können einen vorhandenen OSS serverlosen Redis-Cache mithilfe der Konsole auf die Valkey-Engine aktualisieren CLI, API oder indem Sie die neue Engine und die Hauptversion der Engine mit dem angeben. `modify-serverless-cache` API

## Für Linux, macOS oder Unix:

```
aws elasticache modify-serverless-cache \
 --serverless-cache-name myCluster \
 --engine valkey \
 --major-engine-version 7
```

## Für Windows:

```
aws elasticache modify-serverless-cache ^
 --serverless-cache-name myCluster ^
 --engine valkey ^
 --major-engine-version 7
```

## Unterstützte Engines und Versionen

ElastiCache Serverlose Caches unterstützen Valkey 7.2 und höher, Redis OSS Version 7.0 und Memcached 1.6 und höher.

ElastiCache selbst entworfene Caches unterstützen Valkey 7.2 und höher, alle OSS Redis-Versionen 4.0.10 und höher sowie Memcached-Versionen 1.4.5 und höher.

Selbst entworfene ElastiCache Cluster unterstützen die folgenden Valkey-Versionen:

- [Unterstützte Valkey-Versionen](#)
- [Unterstützte Redis-Versionen OSS](#)
- [Zeitplan für das Ende des Lebenszyklus der Redis-Versionen OSS](#)
- [Unterstützte ElastiCache \(Memcached\) Versionen](#)

## Unterstützte Valkey-Versionen

Nachfolgend finden Sie die unterstützten Valkey-Versionen. Beachten Sie, dass Valkey die meisten in Redis OSS 7.2 verfügbaren Funktionen standardmäßig unterstützt.

### ElastiCache (Valkey) Version 7.2.6

Am 10. Oktober 2024 wurde ElastiCache mit Valkey 7.2.6 veröffentlicht. Hier sind einige der neuen Funktionen, die in Valkey 7.2 eingeführt wurden (im Vergleich zu OSS Redis 7.1):

- Neue WITHSCORE Optionen für ZRANK und Befehle ZREVRANK
- CLIENTNEIN — TOUCH damit Clients Befehle ausführen können, ohneLRU/LFU von Schlüsseln zu beeinflussen.
- Neuer Befehl CLUSTERMYSHARDID, der die Shard-ID des Knotens zurückgibt, um Knoten im Clustermodus auf der Grundlage der Replikation logisch zu gruppieren.
- Leistungs- und Speicheroptimierungen für verschiedene Datentypen.

Hier sind die potenziell schwerwiegenden Verhaltensänderungen zwischen Valkey 7.2 und Redis OSS 7.1 (oder 7.0):

- Wenn Sie PUBLISH mit einem RESP3 Kunden anrufen, der auch denselben Kanal abonniert hat, wird die Reihenfolge geändert und die Antwort wird vor der veröffentlichten Nachricht gesendet.
- Das clientseitige Tracking für Skripte verfolgt jetzt die Schlüssel, die vom Skript gelesen werden, und nicht die Schlüssel, die vom Aufrufer von EVAL/deklariert wurden. FCALL
- Das Freeze Time-Sampling erfolgt während der Befehlsausführung und in Skripten.
- Wenn ein blockierter Befehl entsperrt wird, werden Prüfungen wie ACLOOM, und andere erneut ausgewertet.

- ACLDer Text der Fehlermeldung und die Fehlercodes sind vereinheitlicht.
- Ein blockierter Stream-Befehl, der freigegeben wird, wenn der Schlüssel nicht mehr existiert, enthält einen anderen Fehlercode (- NOGROUP oder - WRONGTYPE statt -UNBLOCKED).
- Die Befehlsstatistiken für blockierte Befehle werden nur aktualisiert, wenn der Befehl tatsächlich ausgeführt wird.
- Durch den internen Speicher von ACL Benutzern werden redundante Befehls- und Kategorieregeln nicht mehr entfernt. Dadurch kann sich die Art und Weise ändern, wie diese Regeln als Teil von ACLSAVE, ACL GETUSER und angezeigt werden ACLLIST.
- Verwenden SNI Sie nach Möglichkeit alle Client-Verbindungen, die für die TLS basierte Replikation erstellt wurden.
- XINFOSTREAM: Das Antwortfeld „Sendezeit“ steht jetzt für die letzte versuchte Interaktion und nicht mehr für die letzte erfolgreiche Interaktion. Das neue Antwortfeld zur aktiven Zeit kennzeichnet jetzt die letzte erfolgreiche Interaktion.
- XREADGROUPund X [AUTO] CLAIM erzeugen den Verbraucher, unabhängig davon, ob er in der Lage war, etwas zu lesen oder einen Anspruch geltend zu machen. [TBD- was ist das „es“ hier?]
- ACLStandardmäßig setzt neu erstellter Benutzer das Sanitize-Payload-Flag in/. ACL LIST GETUSER
- Der HELLO Befehl hat keinen Einfluss auf den Client-Status, sofern er nicht erfolgreich ist.
- NANAntworten werden auf einen einzigen Nan-Typ normalisiert, ähnlich dem aktuellen Verhalten von inf.

[Weitere Informationen zu Valkey finden Sie unter Valkey](#)

[Weitere Informationen zur Version Valkey 7.2 finden Sie in den Versionshinweisen zu Redis OSS 7.2.4 \(Valkey 7.2 enthält alle Änderungen von Redis OSS bis Version 7.2.4\) und in den Versionshinweisen zu Valkey 7.2 unter Valkey am. GitHub](#)

## Unterstützte Redis-Versionen OSS

ElastiCache Serverlose Caches und selbst entworfene Caches unterstützen alle OSS Redis-Versionen 7.1 und früher.

- [ElastiCache \(Valkey\) Version 7.2.6](#)
- [ElastiCache \(RedisOSS\) Version 7.1 \(erweitert\)](#)

Selbst entworfene ElastiCache Cluster unterstützen die folgenden Valkey- und Redis-Versionen: OSS

- [ElastiCache \(RedisOSS\) Version 7.1 \(erweitert\)](#)
- [ElastiCache \(RedisOSS\) Version 7.0 \(erweitert\)](#)
- [ElastiCache \(RedisOSS\) Version 6.2 \(erweitert\)](#)
- [ElastiCache \(RedisOSS\) Version 6.0 \(erweitert\)](#)
- [ElastiCache \(RedisOSS\) Version 5.0.6 \(erweitert\)](#)
- [ElastiCache \(RedisOSS\) Version 5.0.5 \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) Version 5.0.4 \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) Version 5.0.3 \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) Version 5.0.0 \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [ElastiCache \(RedisOSS\) Version 4.0.10 \(erweitert\)](#)
- [Frühere End of Life \(EOL\) -Versionen \(3.x\)](#)
- [Frühere End of Life \(EOL\) -Versionen \(2.x\)](#)

### ElastiCache (RedisOSS) Version 7.1 (erweitert)

Diese Version enthält Leistungsverbesserungen, die es Workloads ermöglichen, einen höheren Durchsatz und geringere Betriebslatenzen zu erzielen. ElastiCache 7.1 führt [zwei Hauptverbesserungen](#) ein:

Wir haben die erweiterte E/A-Thread-Funktionalität auf die Logik der Präsentationsebene ausgedehnt. Mit Darstellungsschicht meinen wir die erweiterten I/O-Threads, die jetzt nicht nur Client-Eingaben lesen, sondern die Eingabe auch in das OSS Redis-Binärbefehlsformat parsen. Diese wird dann zur Ausführung an den Haupt-Thread weitergeleitet, was zu einer Leistungssteigerung führt. Verbessertes OSS Redis-Speicherzugriffsmuster. Die Ausführungsschritte vieler Datenstrukturoperationen sind miteinander verwoben, um Speicherzugriff und reduzierte Speicherzugriffslatenz parallel zu gewährleisten. Kunden, die ElastiCache auf Graviton3 R7g.4xlarge oder höher basieren, können über 1 Million Anfragen pro Sekunde und Knoten erreichen. Mit den Leistungsverbesserungen von ElastiCache (RedisOSS) v7.1 können Kunden im Vergleich zu (Redis) v7.0 einen um bis zu 100% höheren Durchsatz und eine um 50% geringere P99-Latenz erzielen. ElastiCache OSS Diese Verbesserungen sind für Knotengrößen mit mindestens 8 physischen Kernen (2xlarge auf Graviton und 4xlarge auf x86) unabhängig vom CPU Typ aktiviert und erfordern keine Änderungen am Client.

**Note**

ElastiCache v7.1 ist mit OSS Redis v7.0 kompatibel.

## ElastiCache (RedisOSS) Version 7.0 (erweitert)

ElastiCache (RedisOSS) 7.0 bietet eine Reihe von Verbesserungen und Unterstützung für neue Funktionen:

- **[Funktionen](#)**: ElastiCache (RedisOSS) 7 bietet Unterstützung für OSS Redis-Funktionen und bietet eine verwaltete Oberfläche, die es Entwicklern ermöglicht, [LUASkripts](#) mit auf dem ElastiCache Cluster gespeicherter Anwendungslogik auszuführen, ohne dass Clients die Skripts bei jeder Verbindung erneut an den Server senden müssen.
- **[ACLVerbesserungen](#)**: Valkey und Redis OSS 7 bieten Unterstützung für die nächste Version von Access Control Lists (). ACLs Kunden können jetzt mehrere Berechtigungssätze für bestimmte Schlüssel oder Schlüsselräume in Valkey und Redis angeben. OSS
- **[Sharded Pub/Sub](#)**: ElastiCache Mit Valkey und Redis bietet OSS 7 Unterstützung für die gemeinsame Ausführung der Pub/Sub-Funktionalität, wenn sie im Cluster-Modus aktiviert () ausgeführt wird. ElastiCache CME Pub/Sub-Funktionen ermöglichen es Publishern, Nachrichten an eine beliebige Anzahl von Abonnenten auf einem Kanal zu senden. Kanäle sind an einen Shard im ElastiCache Cluster gebunden, sodass Kanalinformationen nicht mehr über mehrere Shards verteilt werden müssen, was zu einer verbesserten Skalierbarkeit führt.
- **Verbessertes I/O-Multiplexing**: ElastiCache Mit Valkey und Redis OSS 7 wurde ein erweitertes I/O-Multiplexing eingeführt, das einen höheren Durchsatz und eine geringere Latenz für Workloads mit hohem Durchsatz und vielen gleichzeitigen Client-Verbindungen zu einem Cluster bietet. ElastiCache Wenn Sie beispielsweise einen Cluster von r6g.xlarge-Knoten verwenden und 5200 Clients gleichzeitig ausführen, können Sie im Vergleich zu Redis OSS Version 6 einen um bis zu 72% höheren Durchsatz (Lese- und Schreibvorgänge pro Sekunde) und eine um bis zu 71% verringerte P99-Latenz erzielen. ElastiCache

[Weitere Informationen zu Valkey finden Sie unter Valkey](#). Weitere Informationen zur Version Redis OSS 7.0 finden Sie in den [Versionshinweisen zu Redis OSS 7.0 unter Redis](#) am. OSS GitHub

## ElastiCache (RedisOSS) Version 6.2 (erweitert)

ElastiCache (RedisOSS) 6.2 beinhaltet Leistungsverbesserungen für TLS -fähige Cluster, die x86-Knotentypen mit 8 vCPUs oder mehr oder Graviton2-Knotentypen mit 4 oder mehr verwenden. vCPUs Diese Verbesserungen verbessern den Durchsatz und reduzieren die Zeit für den Verbindungsaufbau der Clients, indem die Verschlüsselung auf andere übertragen wird. vCPUs Mit Redis OSS 6.2 können Sie auch den Zugriff auf Pub/Sub-Kanäle mithilfe von Regeln für die Zugriffskontrollliste () verwalten. ACL

Mit dieser Version führen wir auch Unterstützung für Daten-Tiering auf Clusterknoten ein, die lokal angeschlossen sind. NVMe SSD Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

Die OSS Redis-Engine-Version 6.2.6 bietet auch Unterstützung für JavaScript das native Format Object Notation (JSON), eine einfache, schemalose Methode zur Kodierung komplexer Datensätze innerhalb von Redis-Clustern. OSS Mit der JSON Unterstützung können Sie die Leistung und Redis für Anwendungen nutzen, die über dem Internet laufen. OSS APIs JSON Weitere Informationen finden Sie unter [Erste Schritte mit JSON](#). Ebenfalls enthalten sind JSON verwandte Metriken `JsonBasedCmds` und `JsonBasedCmdsLatency`, die in die Überwachung der Verwendung dieses Datentyps integriert CloudWatch sind. Weitere Informationen finden Sie unter [Metriken für Valkey und Redis OSS](#).

Sie geben die Engine-Version mit 6.2 an. ElastiCache (RedisOSS) ruft automatisch die bevorzugte Patch-Version von Redis OSS 6.2 auf, die verfügbar ist. Wenn Sie beispielsweise einen Cache-Cluster erstellen/ändern, legen Sie den Parameter `--engine-version` auf 6.2 fest. Der Cluster wird zum Zeitpunkt der Erstellung/Änderung mit der aktuell verfügbaren bevorzugten Patch-Version von Redis OSS 6.2 gestartet. Wenn Sie Engine-Version 6.x in angeben, API erhalten Sie die neueste Nebenversion von Redis 6. OSS

Für bestehende 6.0-Cluster können Sie sich für das nächste Autominor-Versions-Upgrade entscheiden, indem Sie den `AutoMinorVersionUpgrade` Parameter `yes` im `CreateCacheCluster`, `ModifyCacheCluster`, `CreateReplicationGroup` oder `ModifyReplicationGroup` APIs auf setzen. ElastiCache (RedisOSS) aktualisiert die Nebenversion Ihrer vorhandenen 6.0-Cluster mithilfe von Self-Service-Updates auf 6.2. Weitere Informationen finden Sie unter [Self-Service-Updates in Amazon ElastiCache](#).

Beim Aufrufen von wird der `EngineVersion` Parameterwert auf 6.2 gesetzt und die tatsächliche Engine-Version mit der Patch-Version wird im `CacheEngineVersionDescription` Feld zurückgegeben. `DescribeCacheEngineVersions` API

Weitere Informationen zur Version Redis OSS 6.2 finden Sie in den [Versionshinweisen zu Redis OSS 6.2](#) unter OSS Redis on. GitHub

### ElastiCache (RedisOSS) Version 6.0 (erweitert)

Amazon ElastiCache (RedisOSS) stellt die nächste Version der OSS Redis-Engine vor, die die [Authentifizierung von Benutzern mit rollenbasierter Zugriffskontrolle](#), clientseitiges Caching und erhebliche betriebliche Verbesserungen umfasst.

Ab Redis OSS 6.0 wird ElastiCache (RedisOSS) eine einzige Version für jede Redis-Nebenversion anbieten, anstatt mehrere Patch-Versionen anzubieten. ElastiCache (RedisOSS) verwaltet automatisch die Patch-Version Ihrer laufenden Cache-Cluster und sorgt so für eine verbesserte Leistung und erhöhte Sicherheit.

Sie können sich auch für das nächste auto Upgrade der Nebenversion anmelden, indem Sie den `AutoMinorVersionUpgrade` Parameter auf `yes` setzen, und ElastiCache (RedisOSS) verwaltet das Upgrade der Nebenversion über Self-Service-Updates. Weitere Informationen finden Sie unter [Service-Updates in ElastiCache](#).

Sie geben die Engine-Version an, indem Sie `--engine-version 6.0` ElastiCache (RedisOSS) ruft automatisch die bevorzugte Patch-Version von Redis OSS 6.0 auf, die verfügbar ist. Wenn Sie beispielsweise einen Cache-Cluster erstellen/ändern, legen Sie den Parameter `--engine-version` auf `6.0` fest. Der Cluster wird zum Zeitpunkt der Erstellung/Änderung mit der aktuell verfügbaren bevorzugten Patch-Version von Redis OSS 6.0 gestartet. Jede Anfrage mit einem bestimmten Patch-Versionswert wird abgelehnt, eine Ausnahme wird ausgelöst und der Vorgang schlägt fehl.

Beim Aufrufen von `DescribeCacheEngineVersions` wird der `EngineVersion` Parameterwert auf `6.0` gesetzt und die tatsächliche Engine-Version mit der Patch-Version wird im Feld `CacheEngineVersionDescription` zurückgegeben.

Weitere Informationen zur Version Redis OSS 6.0 finden Sie in den [Versionshinweisen zu Redis OSS 6.0](#) unter OSS Redis on. GitHub

### ElastiCache (RedisOSS) Version 5.0.6 (erweitert)

Amazon ElastiCache (RedisOSS) stellt die nächste Version der OSS Redis-Engine vor, die Fehlerkorrekturen und die folgenden kumulativen Updates enthält:

- Garantierte Engine-Stabilität in speziellen Bedingungen.
- Verbesserte Hyperloglog-Fehlerbehandlung.

- Erweiterte Handshake-Befehle für eine zuverlässige Replikation.
- Konsistente Nachverfolgung der Nachrichtenzustellung über den XCLAIM-Befehl.
- Verbesserte LFU -Feldverwaltung in Objekten.
- Verbesserte Transaktionsverwaltung bei Verwendung von ZPOP.
- Möglichkeit, Befehle umzubenennen: Ein Parameter `rename-commands`, mit dem Sie potenziell gefährliche oder teure OSS Redis-Befehle umbenennen können, die zu versehentlichem Datenverlust führen könnten, wie z. B. `FLUSHALL` `FLUSHDB`. Dies ähnelt der `Rename-Command-Konfiguration` in Open-Source-Redis. OSS ElastiCache hat jedoch die Benutzererfahrung durch die Bereitstellung eines vollständig verwalteten Workflows verbessert. Die Befehlsnamensänderungen werden sofort übernommen und automatisch in allen Knoten im Cluster vorgenommen, die die Befehlsliste enthalten. Es ist kein Eingriff Ihrerseits erforderlich, wie etwa das Neustarten der Knoten.

Die folgenden Beispiele zeigen, wie Sie vorhandene Parametergruppen ändern. Sie enthalten den `rename-commands`-Parameter, eine durch Leerzeichen getrennte Liste von Befehlen, die Sie umbenennen möchten:

```
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 diesem Beispiel wird der `rename-commands`-Parameter zum Umbenennen des `flushall`-Befehls in `restrictedflushall` verwendet.

Zum Umbenennen mehrerer Befehle verwenden Sie Folgendes:

```
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
```

Zum Zurücksetzen einer Änderung führen Sie den Befehl erneut aus und schließen sämtliche umbenannten Werte folgendermaßen von der `ParameterValue`-Liste aus, die Sie beibehalten möchten:

```
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 diesem Fall wird der `flushall`-Befehl in `restrictedflushall` umbenannt und sämtliche anderen umbenannten Befehle werden auf ihre ursprünglichen Befehlsnamen zurückgesetzt.

### Note

Beim Umbenennen von Befehlen unterliegen Sie den folgenden Einschränkungen:

- Alle umbenannten Befehle müssen alphanumerisch sein.
- Die maximale Länge neuer Befehlsnamen beträgt 20 alphanumerische Zeichen.
- Beim Umbenennen von Befehlen muss sichergestellt werden, dass Sie die Parametergruppe aktualisieren, die mit Ihrem Cluster verknüpft ist.
- Um die Nutzung eines Befehls gänzlich zu verhindern, verwenden Sie das Schlüsselwort `blocked` folgendermaßen:

```
aws elasticache modify-cache-parameter-group --cache-parameter-group-name custom_param_group --parameter-name-values "ParameterName=rename-commands, ParameterValue='flushall blocked'" --region region
```

Weitere Informationen zu den Parameteränderungen und eine Liste der Befehle, die umbenannt werden können, finden Sie unter [Änderungen der Parameter in Redis 5.0.3 OSS](#).

- Redis OSS Streams: Dabei wird eine Protokollstruktur modelliert, die es Produzenten ermöglicht, neue Elemente in Echtzeit anzuhängen. Nachrichten können auch blockierend oder blockierungsfrei abgerufen werden. Streams ermöglichen auch Konsumentengruppen als Gruppe von Clients, die zusammen unterschiedliche Teile desselben Nachrichten-Streams abrufen, ähnlich wie bei [Apache Kafka](#). [Weitere Informationen finden Sie unter Streams](#).
- Unterstützung für eine Reihe von Stream-Befehlen wie etwa `XADD`, `XRANGE` und `XREAD`. Weitere Informationen finden Sie unter [Streams-Befehle](#).
- Eine Anzahl neuer und umbenannter Parameter. Weitere Informationen finden Sie unter [Änderungen der Parameter in Redis 5.0.0 OSS](#).
- Eine neue OSS Redis-Metrik, `StreamBasedCmds`.
- Etwas schnellere Snapshot-Zeit für Redis-KnotenOSS.

**⚠ Important**

ElastiCache (RedisOSS) hat zwei wichtige Bugfixes aus der [OSSRedis-Open-Source-Version 5.0.1](#) zurückportiert. Sie sind im Folgenden aufgeführt:

- RESTOREAntwort nicht übereinstimmend, wenn bestimmte Schlüssel bereits abgelaufen sind.
- Der Befehl XCLAIM kann einen falschen Eintrag zurückgeben oder die Synchronisierung des Protokolls verhindern.

Beide Bugfixes sind in der ElastiCache (Redis-OSS) Unterstützung für die OSS Redis-Engine-Version 5.0.0 enthalten und werden in future Versionsupdates verwendet.

Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 5.0.6](#) auf Redis am. OSS GitHub

ElastiCache (RedisOSS) Version 5.0.5 (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache (RedisOSS) stellt die nächste Version der OSS Redis-Engine vor;. Es beinhaltet Online-Konfigurationsänderungen für ElastiCache (RedisOSS) von Auto-Failover-Clustern während aller geplanten Operationen. Sie können jetzt Ihren Cluster skalieren, die OSS Redis-Engine-Version aktualisieren und Patches und Wartungsupdates anwenden, während der Cluster online bleibt und weiterhin eingehende Anfragen bearbeitet. Das beinhaltet auch Fehlerbehebungen.

Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 5.0.5](#) unter Redis am. OSS GitHub

ElastiCache (RedisOSS) Version 5.0.4 (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache (RedisOSS) stellt die nächste Version der OSS Redis-Engine vor, die von Amazon unterstützt wird. ElastiCache Die Version enthält die folgenden Erweiterungen:

- Garantierte Engine-Stabilität in speziellen Bedingungen.
- Verbesserte Hyperloglog-Fehlerbehandlung.
- Erweiterte Handshake-Befehle für eine zuverlässige Replikation.
- Konsistente Nachverfolgung der Nachrichtenzustellung über den XCLAIM-Befehl.

- Verbesserte LFU -Feldverwaltung in Objekten.
- Verbesserte Transaktionsverwaltung bei Verwendung von ZPOP.

Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 5.0.4](#) auf Redis am. OSS GitHub

ElastiCache (RedisOSS) Version 5.0.3 (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache (RedisOSS) stellt die nächste Version der von Amazon unterstützten OSS Redis-Engine vor, ElastiCache die Fehlerkorrekturen enthält.

ElastiCache (RedisOSS) Version 5.0.0 (veraltet, verwenden Sie Version 5.0.6)

Amazon ElastiCache (RedisOSS) stellt die nächste Hauptversion der von Amazon unterstützten OSS Redis-Engine vor. ElastiCache ElastiCache (RedisOSS) 5.0.0 bietet Unterstützung für die folgenden Verbesserungen:

- Redis OSS Streams: Dies modelliert eine Protokollstruktur, die es Produzenten ermöglicht, neue Elemente in Echtzeit anzuhängen. Nachrichten können auch blockierend oder blockierungsfrei abgerufen werden. Streams ermöglichen auch Konsumentengruppen als Gruppe von Clients, die zusammen unterschiedliche Teile desselben Nachrichten-Streams abrufen, ähnlich wie bei [Apache Kafka](#). [Weitere Informationen finden Sie unter Streams](#).
- Unterstützung für eine Reihe von Stream-Befehlen wie etwa XADD, XRANGE und XREAD. Weitere Informationen finden Sie unter [Streams-Befehle](#).
- Eine Anzahl neuer und umbenannter Parameter. Weitere Informationen finden Sie unter [Änderungen der Parameter in Redis 5.0.0 OSS](#).
- Eine neue OSS Redis-Metrik, `StreamBasedCmds`.
- Etwas schnellere Snapshot-Zeit für Redis-KnotenOSS.

ElastiCache (RedisOSS) Version 4.0.10 (erweitert)

Amazon ElastiCache (RedisOSS) stellt die nächste Hauptversion der von Amazon unterstützten OSS Redis-Engine vor. ElastiCache ElastiCache (RedisOSS) 4.0.10 unterstützt die folgenden Verbesserungen:

- Sowohl die Online-Clustergrößenänderung als auch die Verschlüsselung in einer einzigen ElastiCache (OSSRedis-) Version. Weitere Informationen finden Sie hier:

- [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
- [Online-Resharding für Valkey oder OSS Redis \(Clustermodus aktiviert\)](#)
- [Datensicherheit bei Amazon ElastiCache](#)
- Eine Anzahl neuer Parameter. Weitere Informationen finden Sie unter [Änderungen der Parameter in Redis OSS 4.0.10](#).
- Unterstützung für eine Reihe von Speicherbefehlen wie etwa MEMORY. Weitere Informationen finden Sie unter [Befehle](#) (suchen nachMEMO).
- Unterstützung für Speicherdefragmentierung, während das System online ist, wodurch eine effizientere Speichernutzung ermöglicht wird und mehr Speicher für Ihre Daten zur Verfügung steht.
- Support für asynchrone Leerungen und Löschungen. ElastiCache (RedisOSS) unterstützt Befehle wie UNLINK FLUSHDB und FLUSHALL zur Ausführung in einem anderen Thread als dem Haupt-Thread. Dies hilft, Leistung und Reaktionszeiten für Ihre Anwendungen zu verbessern, indem Speicher asynchron freigegeben wird.
- Eine neue OSS Redis-Metrik, ActiveDefragHits Weitere Informationen finden Sie unter [Metriken für OSS Redis](#).

Valkey- oder Redis-Benutzer OSS (Clustermodus deaktiviert), die OSS Redis-Version 3.2.10 ausführen, können die Konsole verwenden, um ihre Cluster per Online-Upgrade zu aktualisieren.

Vergleich der Größenänderung von ElastiCache Clustern (RedisOSS) und der Verschlüsselungsunterstützung

| Funktion                                   | 3.2.6 | 3.2.10 | 4.0.10 und höher |
|--------------------------------------------|-------|--------|------------------|
| Online-Größenanpassung eines Clusters *    | Nein  | Ja     | Ja               |
| Verschlüsselung während der Übertragung ** | Ja    | Nein   | Ja               |
| Verschlüsselung im Ruhezustand **          | Ja    | Nein   | Ja               |

\* Hinzufügen, Entfernen und Ausgleichen von Shards.

\* Erforderlich für Fed RAMP HIPAA - und PCI DSS konforme Anwendungen. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

## Frühere End of Life (EOL) -Versionen (3.x)

### ElastiCache (RedisOSS) Version 3.2.10 (erweitert)

Amazon ElastiCache (RedisOSS) stellt die nächste Hauptversion der von Amazon unterstützten OSS Redis-Engine vor. ElastiCache (RedisOSS) 3.2.10 führt die Online-Clustergrößenänderung ein, um Shards zum Cluster hinzuzufügen oder daraus zu entfernen, während der Cluster weiterhin eingehende I/O-Anfragen bearbeitet. ElastiCache (RedisOSS) 3.2.10-Benutzer verfügen über alle Funktionen früherer OSS Redis-Versionen, mit Ausnahme der Fähigkeit, ihre Daten zu verschlüsseln. Diese Möglichkeit ist derzeit nur in Version 3.2.6 verfügbar.

### Vergleich der ElastiCache (OSSRedis-) Versionen 3.2.6 und 3.2.10

| Funktion                                   | 3.2.6 | 3.2.10 |
|--------------------------------------------|-------|--------|
| Online-Größenanpassung eines Clusters *    | Nein  | Ja     |
| Verschlüsselung während der Übertragung ** | Ja    | Nein   |
| Verschlüsselung im Ruhezustand **          | Ja    | Nein   |

\* Hinzufügen, Entfernen und Ausgleichen von Shards.

\* Erforderlich für Fed RAMP - und konforme AnwendungenHIPAA. PCI DSS Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

Weitere Informationen finden Sie hier:

- [Online-Resharding für Valkey oder OSS Redis \(Clustermodus aktiviert\)](#)
- [Online-Größenanpassung von Clustern](#)

### ElastiCache (RedisOSS) Version 3.2.6 (erweitert)

Amazon ElastiCache (RedisOSS) stellt die nächste Hauptversion der von Amazon unterstützten OSS Redis-Engine vor. ElastiCache (RedisOSS) 3.2.6-Benutzer haben alle Funktionen

früherer OSS Redis-Versionen sowie die Möglichkeit, ihre Daten zu verschlüsseln. Weitere Informationen finden Sie hier:

- [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#)
- [Verschlüsselung im Ruhezustand in ElastiCache](#)
- [Konformitätsvalidierung für Amazon ElastiCache](#)

### ElastiCache (Redis) Version 3.2.4 (OSSerweitert)

Amazon ElastiCache (RedisOSS) Version 3.2.4 stellt die nächste Hauptversion der OSS Redis-Engine vor, die von Amazon unterstützt wird. ElastiCache ElastiCache (RedisOSS) 3.2.4-Benutzern stehen alle Funktionen früherer OSS Redis-Versionen sowie die Option zur Ausführung im Cluster-Modus oder Nicht-Cluster-Modus zur Verfügung. Die folgende Tabelle enthält eine Zusammenfassung.

### Vergleich von Redis OSS 3.2.4 ohne Cluster-Modus und Cluster-Modus

| Funktion               | Nicht-Cluster-Modus              | Cluster-Modus                               |
|------------------------|----------------------------------|---------------------------------------------|
| Datenpartitionierung   | Nein                             | Ja                                          |
| Räumliche Indizierung  | Ja                               | Ja                                          |
| Knotentyp ändern       | Ja                               | Ja*                                         |
| Replica-Skalierung     | Ja                               | Ja*                                         |
| Horizontale Skalierung | Nein                             | Ja*                                         |
| Datenbank-Support      | Mehrere                          | Einzel                                      |
| Parametergruppe        | <code>default.redis3.2</code> ** | <code>default.redis3.2.cluster.on</code> ** |

\* Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#)

\*\* Oder davon abgeleitet.

## Hinweise:

- Partitionierung – die Möglichkeit, Ihre Daten auf 2 bis 500 Knotengruppen (Shards) mit Replikationsunterstützung für jede Knotengruppe aufzuteilen.
- Geospatiale Indizierung — Redis OSS 3.2.4 bietet Unterstützung für die Geodatenindizierung über sechs Befehle. GEO Weitere Informationen finden Sie in der OSS GEO Redis\*-Befehlsdokumentation [Befehle: GEO auf der Seite Valkey-Befehle](#) (gefiltert). GEO

Informationen zu zusätzlichen Funktionen von Redis OSS 3 finden Sie in den Versionshinweisen zu [Redis OSS 3.2 und den Versionshinweisen](#) zu [OSSRedis](#) 3.0.

Derzeit ElastiCache verwaltetes Valkey oder Redis OSS (Clustermodus aktiviert) unterstützt die folgenden Redis 3.2-Funktionen nicht: OSS

- Replikatmigration
- Cluster-Ausgleich
- Lua Debugger

ElastiCache deaktiviert die folgenden OSS Redis 3.2-Verwaltungsbefehle:

- `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`

Informationen zu den Redis OSS 3.2.4-Parametern finden Sie unter [Änderungen der Parameter in Redis 3.2.4 OSS](#)

Frühere End of Life (EOL) -Versionen (2.x)

ElastiCache (RedisOSS) Version 2.8.24 (erweitert)

Zu den seit Version 2.8.23 hinzugefügten OSS Redis-Verbesserungen gehören Bugfixes und die Protokollierung fehlerhafter Speicherzugriffsadressen. Weitere Informationen finden Sie in den Versionshinweisen zu [Redis 2.8 OSS](#).

ElastiCache (RedisOSS) Version 2.8.23 (erweitert)

Die seit Version 2.8.22 hinzugefügten OSS Redis-Verbesserungen beinhalten Fehlerkorrekturen. Weitere Informationen finden Sie in den Versionshinweisen zu [Redis 2.8 OSS](#). Diese Version umfasst außerdem Support für den neuen Parameter `close-on-slave-write` der in aktiviertem Zustand die Verbindung zu Clients trennt, die versuchen, in ein schreibgeschütztes Replikat zu schreiben.

Weitere Informationen zu den Redis OSS 2.8.23-Parametern finden Sie [Redis OSS 2.8.23 \(erweitert\) hat Parameter hinzugefügt](#) im Benutzerhandbuch. ElastiCache

ElastiCache (RedisOSS) Version 2.8.22 (erweitert)

Zu den seit Version OSS 2.8.21 hinzugefügten Redis-Verbesserungen gehören:

- Support für unvergabelte Sicherungen und Synchronisationen, sodass Sie weniger Speicher für die Sicherungsverwaltung und mehr für Ihre Anwendung zuweisen können. Weitere Informationen finden Sie unter [So werden Synchronisation und Backup implementiert](#). Der unvergabelte Prozess kann sich sowohl auf die Latenz als auch den Durchsatz auswirken. Bei einem hohen Schreibdurchsatz kann ein Replikat, das erneut synchronisiert wird, für den gesamten Synchronisierungszeitraum nicht erreichbar sein.
- Bei einem Failover werden Replikationsgruppen nun schneller wiederhergestellt, da Replikate möglichst eine partielle Synchronisierung mit dem primären Knoten anstelle von vollständigen Synchronisierungen durchführen. Darüber hinaus verwenden sowohl der primäre Knoten als auch die Replicas während der Synchronisierungen keinen Datenträger mehr, was sich positiv auf die Geschwindigkeit auswirkt.
- Support für zwei neue CloudWatch Metriken.
  - `ReplicationBytes` – Die Anzahl der Bytes, die der primäre Cluster einer Replikationsgruppe an die Lesereplikate sendet.

- `SaveInProgress` – Ein Binärwert, der angibt, ob ein Speicherprozess im Hintergrund ausgeführt wird oder nicht.

Weitere Informationen finden Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#).

- Eine Reihe kritischer Bugfixes im PSYNC Replikationsverhalten. Weitere Informationen finden Sie in den [Versionshinweisen zu Redis OSS 2.8](#).
- Um die verbesserte Replikationsleistung in Multi-AZ-Replikationsgruppen aufrechtzuerhalten und die Cluster-Stabilität zu erhöhen, werden ElastiCache Nicht-Replikate nicht mehr unterstützt.
- Zur Verbesserung der Datenkonsistenz zwischen dem primären Cluster und Replikaten in einer Replikationsgruppe entfernen die Replikate keine Schlüssel unabhängig vom primären Cluster mehr.
- OSSRedis-Konfigurationsvariablen `appendonly` und `appendfsync` werden in Redis OSS Version 2.8.22 und höher nicht unterstützt.
- In Situationen mit geringem Speicher können Clients mit einem großen Ausgabepuffer von einem Replikat-Cluster getrennt werden. In diesem Fall muss der Client die Verbindung erneut herstellen. Solche Situationen treten am wahrscheinlichsten bei Kunden auf. PUBSUB

#### ElastiCache (RedisOSS) Version 2.8.21

Die seit Version 2.8.19 hinzugefügten OSS Redis-Verbesserungen beinhalten eine Reihe von Fehlerkorrekturen. Weitere Informationen finden Sie in den Versionshinweisen zu [Redis 2.8 OSS](#).

#### ElastiCache (RedisOSS) Version 2.8.19

Zu den seit Version OSS 2.8.6 hinzugefügten Redis-Verbesserungen gehören:

- Support für HyperLogLog. Weitere Informationen finden Sie unter [OSSNeue Datenstruktur von Redis: HyperLogLog](#).
- Der Datentyp „Sorted Set“ verfügt mit den neuen Befehlen `ZRANGEBYLEX`, `ZLEXCOUNT` und `ZREMRANGEBYLEX` nun über Support für lexikografische Bereichsabfragen.
- Um zu verhindern, dass ein primärer Knoten veraltete Daten an Replikatknoten sendet, schlägt der Master SYNC fehl, wenn ein untergeordneter Speichervorgang (`bgsave`) im Hintergrund abgebrochen wird.
- Support für die HyperLogLogBasedCommands CloudWatchMetrik. Weitere Informationen finden Sie unter [Metriken für Valkey und Redis OSS](#).

## ElastiCache (RedisOSS) Version 2.8.6

Zu den seit Version OSS 2.6.13 hinzugefügten Redis-Verbesserungen gehören:

- Verbesserte Resilienz und Fehlertoleranz für Read Replicas.
- Support für partielle Synchronisierung.
- Support für eine benutzerdefinierte Mindestanzahl von Read Replicas, die jederzeit verfügbar sein müssen.
- Volle Unterstützung für Pub/Sub – Benachrichtigen von Clients über Ereignisse auf dem Server.
- Automatische Erkennung eines Ausfalls des primären Knotens und eines Failovers Ihres primären auf einen sekundären Knoten.

## ElastiCache (Redis) Version 2.6.13 OSS

Redis OSS Version 2.6.13 war die erste Version von Redis, die von Amazon ElastiCache (Redis) OSS unterstützt wurde. OSS Multi-AZ wird auf Redis 2.6.13 nicht unterstützt. OSS

## Zeitplan für das Ende des Lebenszyklus der Redis-Versionen OSS

In diesem Abschnitt werden die Termine für das Ende der Nutzungsdauer (EOL) für ältere Hauptversionen festgelegt, sobald sie angekündigt werden. Auf diese Weise können Sie Versions- und Upgrade-Entscheidungen für die Zukunft treffen.

### Note

ElastiCache (RedisOSS) -Patch-Versionen von 5.0.0 bis 5.0.5 sind veraltet. Verwenden Sie Version 5.0.6 oder höher.

In der folgenden Tabelle sind jede Version und ihr angekündigtes EOL Datum sowie die empfohlene Upgrade-Zielversion zusammengefasst.

In der Vergangenheit EOL

| Quelle für Unterversionen für ElastiCache                            | Empfohlenes Upgrade-Ziel                                                                                                                                                                                                                                                                                                                                                                  | EOLDatum        |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| V 3.2.4, 3.2.6 und 3.2.10<br>3                                       | Version 6.2 oder höher<br><br><div data-bbox="613 470 1045 884" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> <b>Note</b></p> <p>Für die Regionen US-ISO-EAST US-ISO-WEST -1, US-ISOB-EAST -1 und -1 empfehlen wir 5.0.6 oder höher.</p> </div>    | 31. Juli 2023   |
| V 2.8.24, 2.8.23, 2.8.22, 2.8.21,<br>2 2.8.19, 2.8.12, 2.8.6, 2.6.13 | Version 6.2 oder höher<br><br><div data-bbox="613 995 1045 1409" style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> <b>Note</b></p> <p>Für die Regionen -1, -1 und -1 empfehlen wir 5.0.6 oder höher. US-ISO-EAST US-ISO-WEST US-ISOB-EAST</p> </div> | 13. Januar 2023 |

## Unterstützte ElastiCache (Memcached) Versionen

ElastiCache unterstützt die folgenden Memcached-Versionen und aktualisiert auf neuere Versionen. Achten Sie beim Upgrade auf eine neuere Version sorgfältig auf die Bedingungen, die Ihr Upgrade fehlschlagen lassen, wenn sie nicht erfüllt werden.

### ElastiCache für Memcached-Versionen

- [Memcached-Version 1.6.17](#)

- [Memcached-Version 1.6.17](#)
- [Memcached-Version 1.6.12](#)
- [Memcached-Version 1.6.6](#)
- [Memcached-Version 1.5.16](#)
- [Memcached-Version 1.5.10](#)
- [Memcached-Version 1.4.34](#)
- [Memcached-Version 1.4.33](#)
- [Memcached-Version 1.4.24](#)
- [Memcached-Version 1.4.14](#)
- [Memcached-Version 1.4.5](#)

### Memcached-Version 1.6.17

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.22 hinzu. Es sind keine neuen Funktionen enthalten, aber Fehlerbehebungen und kumulative Updates von [Memcached 1.6.18](#).

[Weitere Informationen finden Sie unter 1622 bei Memcached am. ReleaseNotes](#) GitHub

### Memcached-Version 1.6.17

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.17 hinzu. Es sind keine neuen Funktionen enthalten, aber Fehlerbehebungen und kumulative Updates von [Memcached 1.6.17](#).

[Weitere Informationen finden Sie unter 1617 bei Memcached am. ReleaseNotes](#) GitHub

### Memcached-Version 1.6.12

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.12 und Verschlüsselung bei der Übertragung hinzu. Es enthält Fehlerkorrekturen und kumulative Aktualisierungen von [Memcached 1.5.16](#).

[Weitere Informationen finden Sie unter 1612 bei Memcached am. ReleaseNotes](#) GitHub

## Memcached-Version 1.6.6

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.6 hinzu. [Sie enthält keine neuen Funktionen, jedoch Bugfixes und kumulative Updates von Memcached 1.5.16](#). ElastiCache (Memcached) [bietet keine Unterstützung für Extstore](#).

Weitere Informationen finden Sie in Memcached unter [ReleaseNotes166](#). GitHub

## Memcached-Version 1.5.16

ElastiCache for Memcached fügt Unterstützung für Memcached Version 1.5.16 hinzu. Es sind keine neuen Funktionen enthalten, dafür aber Fehlerbehebungen und kumulative Updates von [Memcached 1.5.14](#) und [Memcached 1.5.15](#).

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.5.16 auf [Memcached](#) am. GitHub

## Memcached-Version 1.5.10

ElastiCache für Memcached unterstützt Version 1.5.10 die folgenden Memcached-Funktionen:

- Automatisierter Slab-Ausgleich
- Schnelleres Suchen von Hash-Tabellen durch murmur3-Algorithmus
- LRUsegmentierter Algorithmus.
- LRUCrawler, um im Hintergrund Speicher zurückzugewinnen.
- `--enable-seccomp`: eine Kompilierungszeitoption

Es werden auch die Parameter `no_modern` und `inline_ascii_resp` eingeführt. Weitere Informationen finden Sie unter [Parameteränderungen für Memcached 1.5.10](#).

Zu den Verbesserungen von Memcached, die seit Memcached Version 1.4.34 ElastiCache hinzugefügt wurden, gehören:

- Kumulative Korrekturen, wie ASCII Multigets, -2017-9951 und Limit-Crawls für. CVE metadumper
- Verbesserte Verbindungsverwaltung durch das Schließen von Verbindungen beim Erreichen der maximalen Verbindungsanzahl
- Verbesserte Verwaltung der Elementgröße bei Elementen, deren Größe 1 MB überschreitet
- Bessere Leistung und Optimierung des Speichermehraufwands durch Reduzierung der Speicheranforderungen pro Element um wenige Bytes

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.5.10 unter [Memcached](#) am. GitHub

### Memcached-Version 1.4.34

ElastiCache für Memcached fügt Version 1.4.34 keine neuen Funktionen zu Version 1.4.33 hinzu. Version 1.4.34 dient der Fehlerbehebung und ist größer als für eine solche Version üblich.

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.4.34 auf [Memcached](#) am. GitHub

### Memcached-Version 1.4.33

Es wurden folgende Memcached-Verbesserungen seit Version 1.4.24 eingeführt:

- Die Fähigkeit, alle Metadaten für eine bestimmte Slab-Klasse, eine Liste der Slab-Klassen oder alle Slab-Klassen zu sichern. Weitere Informationen finden Sie unter [Memcached 1.4.31 Release Notes](#).
- Verbesserter Support für große Elemente über den Standardwert von 1 MB. Weitere Informationen finden Sie unter [Memcached 1.4.29 Release Notes](#).
- Die Fähigkeit, anzugeben, wie lange sich ein Client im Leerlauf befinden kann, bevor er zum Schließen aufgefordert wird.

Die Fähigkeit, den für Memcached verfügbaren Speicherplatz dynamisch zu erhöhen, ohne den Cluster neu starten zu müssen. Weitere Informationen finden Sie unter [Memcached 1.4.27 Release Notes](#).

- Die Protokollierung von `fetchers`, `mutations` und `evictions` wird jetzt unterstützt. Weitere Informationen finden Sie unter [Memcached 1.4.26 Release Notes](#).
- Freier Speicher kann für einen globalen Pool wieder freigegeben und neuen Slab-Klassen zugewiesen werden. Weitere Informationen finden Sie unter [Memcached 1.4.25 Release Notes](#).
- Mehrere Fehlerbehebungen
- Einige neue Befehle und Parameter Eine Liste finden Sie hier: [In Memcached 1.4.33 hinzugefügte Parameter](#).

### Memcached-Version 1.4.24

Es wurden folgende Memcached-Verbesserungen seit Version 1.4.14 eingeführt:

- In letzter Zeit verwendete Verwaltung (LRU) mithilfe eines Hintergrundprozesses.
- Die Option, jenkins oder murmur3 als Hash-Algorithmus zu verwenden, wurde hinzugefügt.
- Einige neue Befehle und Parameter Eine Liste finden Sie hier: [In Memcached 1.4.24 hinzugefügte Parameter](#).
- Mehrere Fehlerbehebungen

#### Memcached-Version 1.4.14

Es wurden folgende Memcached-Verbesserungen seit Version 1.4.5 eingeführt:

- Verbesserter Slab-Ausgleich.
- Performance und Skalierbarkeit wurden verbessert.
- Der Befehl touch wurde eingeführt, um das Ablaufdatum eines vorhandenen Elements zu aktualisieren, ohne es abzurufen.
- Automatische Erkennung – die Möglichkeit für Client-Programme, automatisch alle Cache-Knoten in einem Cluster zu ermitteln und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten.

#### Memcached-Version 1.4.5

Memcached Version 1.4.5 war die erste Engine und Version, die von Amazon unterstützt wurde ElastiCache (Memcached).

## Wesentliche Unterschiede im Versionsverhalten und in der Kompatibilität mit Valkey

Valkey 7.2.6 weist ähnliche Kompatibilitätsunterschiede zu früheren Versionen von Redis 7.2.5 auf. OSS Die neueste unterstützte Version von Valkey finden Sie unter. [Unterstützte Engines und Versionen](#)

## Wesentliche Unterschiede im Versionsverhalten und in der Kompatibilität mit Redis OSS

### Important

Auf der folgenden Seite sind alle Inkompatibilitätsunterschiede zwischen den Versionen aufgeführt und alle Aspekte angegeben, die Sie beim Upgrade auf neuere Versionen beachten sollten. In dieser Liste sind alle Probleme bezüglich Versionsinkompatibilitäten aufgeführt, die beim Upgrade auftreten können.

Sie können direkt von Ihrer aktuellen OSS Redis-Version auf die neueste verfügbare Redis-Version aktualisieren, ohne dass OSS sequentielle Upgrades erforderlich sind. Sie können beispielsweise direkt von Redis OSS Version 3.0 auf Version 7.0 aktualisieren.

OSSRedis-Versionen werden mit einer semantischen Version identifiziert, die eine MAJORMINOR, und -Komponente umfasst. PATCH In Redis OSS 4.0.10 ist die Hauptversion beispielsweise 4, die Nebenversion 0 und die Patch-Version 10. Diese Werte werden im Allgemeinen basierend auf den folgenden Konventionen schrittweise erhöht:

- MAJORVersionen sind für inkompatible Änderungen API
- MINORVersionen sind für neue Funktionen vorgesehen, die abwärtskompatibel hinzugefügt wurden
- PATCHVersionen sind für abwärtskompatible Bugfixes und nicht funktionale Änderungen vorgesehen

Wir empfehlen, innerhalb eines bestimmten Zeitraums immer auf der neuesten Patch-Version zu bleiben. MAJOR MINORVersion, um die neuesten Leistungs- und Stabilitätsverbesserungen zu erzielen. Ab Redis OSS 6.0 wird ElastiCache (RedisOSS) eine einzige Version für jede OSS Redis-Nebenversion anbieten, anstatt mehrere Patch-Versionen anzubieten. ElastiCache (RedisOSS)

verwaltet automatisch die Patch-Version Ihrer laufenden Cache-Cluster und sorgt so für eine verbesserte Leistung und erhöhte Sicherheit.

Wir empfehlen außerdem, regelmäßig auf die neueste Hauptversion zu aktualisieren, da die meisten wichtigen Verbesserungen nicht auf ältere Versionen zurückportiert werden. Da die Verfügbarkeit auf eine neue AWS Region ElastiCache ausgedehnt wird, unterstützt ElastiCache (RedisOSS) die beiden neuesten MAJOR MINOR-Versionen zu diesem Zeitpunkt für die neue Region. Zum Beispiel, wenn eine neue AWS Region eingeführt wird und die neueste MAJOR MINOR ElastiCache (RedisOSS) Versionen sind 7.0 und 6.2, ElastiCache (RedisOSS) wird die Versionen 7.0 und 6.2 in der neuen AWS Region unterstützen. Wie neuer MAJOR MINOR-Versionen von ElastiCache (RedisOSS) wurden veröffentlicht. Es ElastiCache wird weiterhin Unterstützung für die neu veröffentlichten ElastiCache (Redis-OSS) Versionen hinzugefügt. Weitere Informationen zur Auswahl von Regionen für ElastiCache finden Sie unter [Regionen und Verfügbarkeitszonen auswählen](#).

Wenn Sie ein Upgrade durchführen, das Haupt- oder Nebenversionen umfasst, beachten Sie bitte die folgende Liste, die Verhaltens- und rückwärtsinkompatible Änderungen enthält, die im Laufe der Zeit mit Redis OSS veröffentlicht wurden.

## Verhalten und abwärtsinkompatible Änderungen von Redis OSS 7.0

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 7.0](#).

- SCRIPT LOAD und SCRIPT FLUSH werden nicht mehr auf Replikate weitergeleitet. Wenn Sie eine gewisse Haltbarkeit von Skripten benötigen, empfehlen wir Ihnen, die Verwendung von [OSSRedis-Funktionen](#) in Betracht zu ziehen.
- Pubsub-Kanäle sind jetzt standardmäßig für neue ACL Benutzer gesperrt.
- Der STRALGO-Befehl wurde durch den LCS-Befehl ersetzt.
- Das Format für ACL GETUSER wurde geändert, sodass alle Felder das standardmäßige Zugriffszeichenfolgemuster anzeigen. Wenn Sie Automatisierung mit ACL GETUSER verwendet haben, sollten Sie überprüfen, ob beide Formate verarbeitet werden.
- Die ACL Kategorien für SELECT, WAIT, ROLE, LASTSAVE READONLY READWRITE, und ASKING haben sich geändert.
- Der INFO-Befehl zeigt jetzt Befehlsstatistiken pro Unterbefehl und anstatt bei den Container-Befehlen in der obersten Ebene an.
- Die Rückgabewerte der Befehle LPOP, RPOP, ZPOPMIN und ZPOPMAX haben sich in bestimmten Randfällen geändert. Wenn Sie diese Befehle verwenden, sollten Sie die Versionshinweise überprüfen und bewerten, ob Sie betroffen sind.

- Die Befehle SORT und SORT\_RO erfordern jetzt Zugriff auf den gesamten Schlüsselraum, um die Argumente GET sowie BY verwenden zu können.

## Verhalten und abwärtsinkompatible Änderungen in Redis OSS 6.2

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 6.2](#).

- Die ACL Flags der LASTSAVE Befehle TIMEECHO,ROLE, und wurden geändert. Dies kann dazu führen, dass Befehle, die zuvor gestattet waren, abgelehnt werden und umgekehrt.

### Note

Keiner dieser Befehle verändert Daten oder gewährt Zugriff darauf.

- Beim Upgrade von Redis OSS 6.0 wurde die Reihenfolge der Schlüssel/Wert-Paare, die von einer Map-Antwort an ein Lua-Skript zurückgegeben wurden, geändert. Wenn Ihre Skripte eine Map verwenden `redis.setresp()` oder zurückgeben (neu in Redis OSS 6.0), sollten Sie die Auswirkungen berücksichtigen, die das Skript bei Upgrades unterbrechen kann.

## Verhalten von Redis OSS 6.0 und abwärtsinkompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 6.0](#).

- Die maximale Anzahl zugelassener Datenbanken wurde von 1,2 Millionen auf 10.000 verringert. Der Standardwert ist 16, und wir raten davon ab, Werte zu verwenden, die viel größer sind, da wir Probleme mit der Leistung und dem Arbeitsspeicher festgestellt haben.
- Setzen Sie den `AutoMinorVersionUpgrade` Parameter auf `yes`, und ElastiCache (RedisOSS) verwaltet das Upgrade der Nebenversion über Self-Service-Updates. Dies wird über Standardkanäle für die Kundenbenachrichtigung über eine Self-Service-Update-Kampagne abgewickelt. Weitere Informationen finden Sie unter [Self-Service-Updates](#) unter ElastiCache

## Verhalten und abwärtsinkompatible Änderungen in Redis OSS 5.0

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 5.0](#).

- Skripte werden durch Effekte repliziert, statt das Skript auf dem Replikat erneut auszuführen. Dies verbessert im Allgemeinen die Leistung, kann jedoch die Menge der zwischen primären Replikaten

und Replikaten replizierten Daten erhöhen. Es gibt eine Option, um zum vorherigen Verhalten zurückzukehren, die nur in ElastiCache (RedisOSS) 5.0 verfügbar ist.

- Wenn Sie ein Upgrade von Redis OSS 4.0 durchführen, geben einige Befehle in LUA Skripten Argumente in einer anderen Reihenfolge zurück als in früheren Versionen. In Redis OSS 4.0 ordnete Redis OSS einige Antworten lexographisch an, um die Antworten deterministisch zu machen. Diese Reihenfolge wird nicht angewendet, wenn Skripten durch Effekte repliziert werden.
- In Redis OSS 5.0.3 und höher wird ElastiCache (RedisOSS) bei Instance-Typen mit mehr als 4 einen Teil der I/O-Arbeit auf Hintergrundkerne auslagern. VCPUs Dies kann die Leistungsmerkmale von Redis OSS und die Werte einiger Metriken ändern. Weitere Informationen finden Sie unter [Welche Metriken sollte ich überwachen?](#), um zu verstehen, ob Sie ändern müssen, welche Metriken Sie sich ansehen.

## Verhalten von Redis OSS 4.0 und abwärtsinkompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 4.0](#).

- Slow Log protokolliert jetzt zwei zusätzliche Argumente, den Namen und die Adresse des Clients. Diese Änderung sollte abwärtskompatibel sein, sofern Sie sich nicht explizit darauf verlassen, dass jeder Slow-Log-Eintrag 3 Werte enthält.
- Der Befehl CLUSTER NODES gibt jetzt ein etwas anderes Format zurück, das nicht abwärtskompatibel ist. Wir empfehlen, dass Clients diesen Befehl nicht verwenden, um mehr über die in einem Cluster vorhandenen Knoten zu erfahren, sondern stattdessen CLUSTER SLOTS verwenden.

## In der Vergangenheit EOL

### Verhalten und abwärtsinkompatible Änderungen in Redis OSS 3.2

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 3.2](#).

- Für diese Version sind keine Kompatibilitätsänderungen erforderlich.

Weitere Informationen finden Sie unter [Zeitplan für das Ende des Lebenszyklus der Redis-Versionen OSS](#).

### Verhalten von Redis OSS 2.8 und abwärtsinkompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie in den Versionshinweisen zu [Redis OSS 2.8](#).

- Ab Redis OSS 2.8.22 OSS AOF wird Redis in (Redis) nicht mehr unterstützt. ElastiCache OSS Wir empfehlen die Verwendung von MemoryDB, wenn Daten dauerhaft gespeichert werden müssen.
- Ab Redis OSS 2.8.22 unterstützt ElastiCache (RedisOSS) das Anhängen von Replikaten an darin gehostete Primärdateien nicht mehr. ElastiCache Während des Upgrades werden externe Replikate getrennt und sie können keine erneute Verbindung herstellen. Wir empfehlen die Verwendung von clientseitigem Caching, das in Redis 6.0 als Alternative zu externen Replikaten verfügbar ist. OSS
- Die Befehle TTL und PTTL geben jetzt -2 zurück, wenn der Schlüssel nicht existiert, und -1, wenn er existiert, aber kein zugehöriges Ablaufdatum hat. Redis OSS 2.6 und frühere Versionen gaben für beide Bedingungen -1 zurück.
- SORT mit ALPHA sortiert jetzt nach lokalem Standardgebietschema, wenn keine STORE-Option verwendet wird.

Weitere Informationen finden Sie unter [Zeitplan für das Ende des Lebenszyklus der Redis-Versionen OSS](#).

## Blockierte Valkey- oder OSS Redis-Engine-Upgrades beheben

Wie in der folgenden Tabelle dargestellt, ist Ihr Valkey- oder OSS Redis-Engine-Upgrade-Vorgang blockiert, wenn ein Scale-Up-Vorgang aussteht.

| Ausstehende Vorgänge                  | Blockierte Vorgänge             |
|---------------------------------------|---------------------------------|
| Aufwärtsskalierung                    | Unmittelbares Engine-Upgrade    |
| Engine-Upgrade                        | Unmittelbares Aufwärtsskalieren |
| Aufwärtsskalierung und Engine-Upgrade | Unmittelbares Aufwärtsskalieren |
|                                       | Unmittelbares Engine-Upgrade    |

Um ein blockiertes OSS Redis-Engine-Upgrade zu beheben

- Führen Sie eine der folgenden Aktionen aus:
  - Planen Sie Ihren OSS Redis-Engine-Upgrade-Vorgang für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Sofort anwenden deaktivieren.

Verwenden `--no-apply-immediately` Sie mit CLI dem. Mit demAPI, benutze `ApplyImmediately=false`.

- Warten Sie bis zum nächsten Wartungsfenster (oder danach), um Ihr OSS Redis-Engine-Upgrade durchzuführen.
- Fügen Sie dieser Cluster-Änderung den OSS Redis-Scale-Up-Vorgang hinzu, wobei das Kontrollkästchen Sofort anwenden aktiviert ist.

Verwenden `--apply-immediately` Sie mit CLI dem. Mit demAPI, benutze `ApplyImmediately=true`.

Dieser Ansatz bricht das Engine-Upgrade während des nächsten Wartungsfensters effektiv ab, indem es direkt ausgeführt wird.

## ElastiCache bewährte Methoden und Caching-Strategien

Im Folgenden finden Sie empfohlene Best Practices für Amazon ElastiCache. Durch die Einhaltung dieser Methoden lassen sich die Performance und Zuverlässigkeit des Caches verbessern.

### Themen

- [Allgemeine Best Practices](#)
- [Unterstützte und eingeschränkte Valkey-, Redis OSS - und Memcached-Befehle](#)
- [OSSKonfiguration und Limits von Valkey und Redis](#)
- [IPv6Kundenbeispiele für Valkey, Redis OSS und Memcached](#)
- [Best Practices für Kunden \(Valkey und OSS Redis\)](#)
- [Bewährte Methoden für Kunden \(Memcached\)](#)
- [TLSaktivierte ElastiCache Dual-Stack-Cluster](#)
- [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#)
- [Bewährte Methoden bei der Arbeit mit selbst entworfenen Clustern von Valkey und Redis OSS](#)
- [Caching-Strategien für Memcached](#)

## Allgemeine Best Practices

Im Folgenden finden Sie Informationen zu bewährten Methoden für die Verwendung der darin enthaltenen Valkey-, Redis OSS - und Memcached-Schnittstellen. ElastiCache

- Verwenden Sie Konfigurationen mit aktiviertem Clustermodus — Mit aktiviertem Clustermodus kann der Cache horizontal skaliert werden, um mehr Speicherplatz und Durchsatz zu erzielen als bei einer deaktivierten Konfiguration im Clustermodus. ElastiCache Serverless ist nur in einer Konfiguration mit aktiviertem Clustermodus verfügbar.
- Verwenden Sie langlebige Verbindungen — Das Erstellen einer neuen Verbindung ist teuer und beansprucht Zeit und CPU Ressourcen aus dem Cache. Verwenden Sie Verbindungen nach Möglichkeit wieder (z. B. mit Verbindungspooling), um diese Kosten für viele Befehle zu amortisieren.
- Aus Replikaten lesen — Wenn Sie ElastiCache serverlose Systeme verwenden oder Read Replicas (selbst entworfene Cluster) bereitgestellt haben, können Sie Lesevorgänge direkt an Replikate senden, um eine bessere Skalierbarkeit und/oder eine geringere Latenz zu erreichen. Lesevorgänge aus Replikaten sind mit dem Primärknoten letztendlich konsistent.

Vermeiden Sie es, in einem selbst entworfenen Cluster Leseanforderungen an eine einzelne Read Replica weiterzuleiten, da Lesevorgänge möglicherweise vorübergehend nicht verfügbar sind, wenn der Knoten ausfällt. Konfigurieren Sie Ihren Client entweder so, dass Leseanfragen an mindestens zwei Read Replicas weitergeleitet werden oder Lesevorgänge an ein einzelnes Replikat und den Primärknoten weitergeleitet werden.

Im ElastiCache serverlosen Modus werden beim Lesen vom Replikat-Port (6380) die Lesevorgänge nach Möglichkeit in die lokale Availability Zone des Clients geleitet, wodurch die Latenz beim Abrufen reduziert wird. Bei Ausfällen wird automatisch auf die anderen Knoten zurückgegriffen.

- Vermeiden Sie teure Befehle – Vermeiden Sie die Ausführung rechen- und E/A-intensiver Operationen, wie z. B. die Befehle KEYS und SMEMBERS. Wir empfehlen diesen Ansatz, da diese Operationen die Last auf dem Cluster erhöhen und Einfluss auf die Performance des Clusters haben. Verwenden Sie stattdessen die Befehle SCAN und SSCAN.
- Befolgen Sie die bewährten Methoden von Lua – Vermeiden Sie lange laufende Lua-Skripte und deklarieren Sie Schlüssel, die in Lua-Skripten verwendet werden, immer im Voraus. Wir empfehlen diesen Ansatz, um festzustellen, dass im Lua-Skript keine slotübergreifenden Befehle verwendet werden. Vergewissern Sie sich, dass die in Lua-Skripten verwendeten Schlüssel zum gleichen Slot gehören.
- Sharded Pub/Sub verwenden — Wenn Sie Valkey oder Redis OSS zur Unterstützung von Pub/Sub-Workloads mit hohem Durchsatz verwenden, empfehlen wir die Verwendung von [Sharded Pub/Sub](#) (verfügbar mit Valkey und mit Redis 7 oder höher). OSS Herkömmliche Pub/Sub-Cluster mit aktiviertem Cluster-Modus senden Nachrichten an alle Knoten im Cluster, was zu hoher

EngineCPUUtilization führen kann. Beachten Sie, dass bei serverlosen, herkömmlichen Pub/Sub-Befehlen intern Sharded Pub/Sub-Befehle verwendet werden. ElastiCache

## Unterstützte und eingeschränkte Valkey-, Redis OSS - und Memcached-Befehle

### Unterstützte Valkey- und Redis-Befehle OSS

#### Unterstützte Valkey- und Redis-Befehle OSS

Die folgenden Valkey- und OSS Redis-Befehle werden von serverlosen Caches unterstützt.

Zusätzlich zu diesen Befehlen werden außerdem diese [Unterstützte Valkey- und Redis-Befehle OSS](#) unterstützt.

#### Bitmap-Befehle

- BITCOUNT

Zählt die Anzahl der gesetzten Bits (Populationszählung) in einer Zeichenfolge.

#### [Weitere Informationen](#)

- BITFIELD

Führt willkürliche Bitfeld-Ganzzahl-Operationen für Zeichenfolgen aus.

#### [Weitere Informationen](#)

- BITFIELD\_RO

Führt willkürliche, schreibgeschützte Bitfeld-Ganzzahl-Operationen für Zeichenfolgen aus.

#### [Weitere Informationen](#)

- BITOP

Führt bitweise Operationen an mehreren Zeichenfolgen aus und speichert das Ergebnis.

#### [Weitere Informationen](#)

- BITPOS

Findet das erste gesetzte (1) oder nicht gesetzte Bit (0) in einer Zeichenfolge.

[Weitere Informationen](#)

- GETBIT

Gibt einen Bitwert nach Offset zurück.

[Weitere Informationen](#)

- SETBIT

Setzt oder löscht das Bit am Offset des Zeichenfolgenwerts. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

Befehle zur Clusterverwaltung

- CLUSTER COUNTKEYSINSLOT

Gibt die Anzahl der Schlüssel in einem Hash-Slot zurück.

[Weitere Informationen](#)

- CLUSTER GETKEYSINSLOT

Gibt die Schlüsselnamen in einem Hash-Slot zurück.

[Weitere Informationen](#)

- CLUSTER INFO

Gibt Informationen zum Status eines Knotens zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

[Weitere Informationen](#)

- CLUSTER KEYSLOT

Gibt den Hash-Slot für einen Schlüssel zurück.

[Weitere Informationen](#)

- CLUSTER MYID

Gibt die ID eines Knotens zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

### [Weitere Informationen](#)

- CLUSTER NODES

Gibt die Clusterkonfiguration für einen Knoten zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

### [Weitere Informationen](#)

- CLUSTER REPLICAS

Listet die Replikatknoten eines Hauptknotens auf. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

### [Weitere Informationen](#)

- CLUSTER SHARDS

Gibt die Zuordnung von Cluster-Slots zu Shards zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

### [Weitere Informationen](#)

- CLUSTER SLOTS

Gibt die Zuordnung von Cluster-Slots zu Knoten zurück. Gibt in einem Serverless-Cache den Status des einzelnen virtuellen „Shard“ zurück, der dem Client angezeigt wird.

### [Weitere Informationen](#)

- READONLY

Ermöglicht schreibgeschützte Abfragen für eine Verbindung zu einem Valkey- oder Redis-Cluster-Replikatknoten. OSS

### [Weitere Informationen](#)

- READWRITE

Aktiviert Lese-/Schreibabfragen für eine Verbindung zu einem Valkey- oder Redis-Cluster-Replikatknoten. OSS

## [Weitere Informationen](#)

### Befehle zur Verbindungsverwaltung

- AUTH

Authentifiziert die Verbindung.

## [Weitere Informationen](#)

- CLIENT GETNAME

Gibt den Namen der Verbindung zurück.

## [Weitere Informationen](#)

- CLIENT REPLY

Weist den Server an, ob er auf Befehle antworten soll.

## [Weitere Informationen](#)

- CLIENT SETNAME

Legt den Verbindungsnamen fest.

## [Weitere Informationen](#)

- ECHO

Gibt die angegebene Zeichenfolge zurück.

## [Weitere Informationen](#)

- HELLO

Handshakes mit dem Valkey- oder Redis-Server. OSS

## [Weitere Informationen](#)

- PING

Gibt die Lebhaftigkeitsantwort des Servers zurück.

## [Weitere Informationen](#)

- QUIT

Schließt die Verbindung.

[Weitere Informationen](#)

- RESET

Setzt die Verbindung zurück.

[Weitere Informationen](#)

- SELECT

Ändert die ausgewählte Datenbank.

[Weitere Informationen](#)

## Generische Befehle

- COPY

Kopiert den Wert eines Schlüssels in einen neuen Schlüssel.

[Weitere Informationen](#)

- DEL

Löscht einen oder mehrere Schlüssel.

[Weitere Informationen](#)

- DUMP

Gibt eine serialisierte Darstellung des in einem Schlüssel gespeicherten Werts zurück.

[Weitere Informationen](#)

- EXISTS

Ermittelt, ob ein oder mehrere Schlüssel existieren.

[Weitere Informationen](#)

- EXPIRE

Legt die Ablaufzeit eines Schlüssels in Sekunden fest.

[Weitere Informationen](#)

- EXPIREAT

Legt die Ablaufzeit eines Schlüssels auf einen Unix-Zeitstempel fest.

[Weitere Informationen](#)

- EXPIRETIME

Gibt die Ablaufzeit eines Schlüssels als Unix-Zeitstempel zurück.

[Weitere Informationen](#)

- PERSIST

Entfernt die Ablaufzeit eines Schlüssels.

[Weitere Informationen](#)

- PEXPIRE

Legt die Ablaufzeit eines Schlüssels in Millisekunden fest.

[Weitere Informationen](#)

- PEXPIREAT

Legt die Ablaufzeit eines Schlüssels auf einen Unix-Millisekunden-Zeitstempel fest.

[Weitere Informationen](#)

- PEXPIRETIME

Gibt die Ablaufzeit eines Schlüssels als Unix-Millisekunden-Zeitstempel zurück.

[Weitere Informationen](#)

- PTTL

Gibt die Ablaufzeit eines Schlüssels in Millisekunden zurück.

[Weitere Informationen](#)

- RANDOMKEY

Gibt einen zufälligen Schlüsselnamen aus der Datenbank zurück.

### [Weitere Informationen](#)

- RENAME

Benennt einen Schlüssel um und überschreibt das Ziel.

### [Weitere Informationen](#)

- RENAMENX

Benennt einen Schlüssel nur um, wenn der Name des Zielschlüssels nicht vorhanden ist.

### [Weitere Informationen](#)

- RESTORE

Erstellt einen Schlüssel aus der serialisierten Darstellung eines Werts.

### [Weitere Informationen](#)

- SCAN

Iteriert über die Schlüsselnamen in der Datenbank.

### [Weitere Informationen](#)

- SORT

Sortiert die Elemente in einer Liste, einem Set oder einem sortierten Set und speichert optional das Ergebnis.

### [Weitere Informationen](#)

- SORT\_RO

Gibt die sortierten Elemente einer Liste, eines Sets oder eines sortierten Sets zurück.

### [Weitere Informationen](#)

- TOUCH

Gibt die Anzahl der vorhandenen Schlüssel aus den angegebenen Schlüsseln zurück, nachdem der Zeitpunkt des letzten Zugriffs aktualisiert wurde.

### [Weitere Informationen](#)

- TTL

Gibt die Ablaufzeit eines Schlüssels in Sekunden zurück.

### [Weitere Informationen](#)

- TYPE

Ermittelt den Typ des Wertes, der in einem Schlüssel gespeichert ist.

### [Weitere Informationen](#)

- UNLINK

Löscht asynchron einen oder mehrere Schlüssel.

### [Weitere Informationen](#)

## Geodatenbefehle

- GEOADD

Fügt einem Geodatenindex ein oder mehrere Mitglieder hinzu. Der Schlüssel wird erstellt, wenn er nicht existiert.

### [Weitere Informationen](#)

- GEODIST

Gibt den Abstand zwischen zwei Mitgliedern eines Geodatenindex zurück.

### [Weitere Informationen](#)

- GEOHASH

Gibt Elemente aus einem Geodatenindex als Geohash-Zeichenfolgen zurück.

### [Weitere Informationen](#)

- GEOPOS

Gibt den Längen- und Breitengrad von Elementen aus einem Geodatenindex zurück.

[Weitere Informationen](#)

- GEORADIUS

Fragt einen Geodatenindex nach Elementen ab, die sich innerhalb einer Entfernung von einer Koordinate befinden, und speichert optional das Ergebnis.

[Weitere Informationen](#)

- GEORADIUS\_RO

Gibt Elemente aus einem Geodatenindex zurück, die sich innerhalb einer Entfernung von einer Koordinate befinden.

[Weitere Informationen](#)

- GEORADIUSBYMEMBER

Fragt einen Geodatenindex nach Elementen ab, die sich innerhalb einer Entfernung von einem Element befinden, und speichert optional das Ergebnis.

[Weitere Informationen](#)

- GEORADIUSBYMEMBER\_RO

Gibt Elemente aus einem Geodatenindex zurück, die sich innerhalb einer Entfernung von einem Element befinden.

[Weitere Informationen](#)

- GEOSEARCH

Fragt einen Geodatenindex nach Elementen ab, die sich in einem Bereich eines Felds oder Kreises befinden.

[Weitere Informationen](#)

- GEOSEARCHSTORE

Fragt einen Geodatenindex nach Elementen ab, die sich in einem Bereich eines Felds oder Kreises befinden, und speichert optional das Ergebnis.

[Weitere Informationen](#)

## Hash-Befehle

- HDEL

Löscht ein oder mehrere Felder und deren Werte aus einem Hash. Löscht den Hash, wenn keine Felder mehr vorhanden sind.

[Weitere Informationen](#)

- HEXISTS

Ermittelt, ob ein Feld in einem Hash existiert.

[Weitere Informationen](#)

- HGET

Gibt den Wert eines Felds in einem Hash zurück.

[Weitere Informationen](#)

- HGETALL

Gibt alle Felder und Werte in einem Hash zurück.

[Weitere Informationen](#)

- HINCRBY

Inkrementiert den Ganzzahlwert eines Felds in einem Hash um eine Zahl. Verwendet 0 als Anfangswert, wenn das Feld nicht existiert.

[Weitere Informationen](#)

- HINCRBYFLOAT

Inkrementiert den Gleitkommawert eines Felds um eine Zahl. Verwendet 0 als Anfangswert, wenn das Feld nicht existiert.

[Weitere Informationen](#)

- HKEYS

Gibt alle Felder in einem Hash zurück.

[Weitere Informationen](#)

- HLEN

Gibt die Anzahl der Felder in einem Hash zurück.

[Weitere Informationen](#)

- HMGET

Gibt die Werte aller Felder in einem Hash zurück.

[Weitere Informationen](#)

- HMSET

Legt die Werte mehrerer Felder fest.

[Weitere Informationen](#)

- HRANDFIELD

Gibt ein oder mehrere zufällige Felder aus einem Hash zurück.

[Weitere Informationen](#)

- HSCAN

Iteriert über Felder und Werte eines Hashs.

[Weitere Informationen](#)

- HSET

Erstellt oder ändert den Wert eines Felds in einem Hash.

[Weitere Informationen](#)

- HSETNX

Legt den Wert eines Felds in einem Hash nur fest, wenn das Feld nicht existiert.

[Weitere Informationen](#)

- HSTRLEN

Gibt die Länge des Werts eines Felds zurück.

[Weitere Informationen](#)

- HVALS

Gibt alle Werte in einem Hash zurück.

[Weitere Informationen](#)

## HyperLogLog Befehle

- PFADD

Fügt einem HyperLogLog Schlüssel Elemente hinzu. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- PFCOUNT

Gibt die ungefähre Kardinalität der Menge (n) zurück, die von dem/den HyperLogLog Schlüssel (n) beobachtet wurde.

[Weitere Informationen](#)

- PFMERGE

Führt einen oder mehrere HyperLogLog Werte zu einem einzigen Schlüssel zusammen.

[Weitere Informationen](#)

## Listenbefehle

- BLMOVE

Löscht ein Element aus einer Liste, verschiebt es in eine andere Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element verschoben wurde.

[Weitere Informationen](#)

- BLMPOP

Löscht das erste Element aus einer von mehreren Listen. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

### [Weitere Informationen](#)

- BLPOP

Entfernt das erste Element in einer Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

### [Weitere Informationen](#)

- BRPOP

Entfernt das letzte Element in einer Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

### [Weitere Informationen](#)

- BRPOPLPUSH

Löscht ein Element aus einer Liste, verschiebt es in eine andere Liste und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht die Liste, wenn das letzte Element gelöscht wurde.

### [Weitere Informationen](#)

- LINDEX

Gibt ein Element anhand seines Index aus einer Liste zurück.

### [Weitere Informationen](#)

- LINSERT

Fügt ein Element vor oder nach einem anderen Element in eine Liste ein.

### [Weitere Informationen](#)

- LLEN

Gibt die Länge einer Liste zurück.

### [Weitere Informationen](#)

- LMOVE

Gibt ein Element zurück, nachdem es aus einer Liste entfernt und in eine andere verschoben wurde. Löscht die Liste, wenn das letzte Element verschoben wurde.

[Weitere Informationen](#)

- LMPOP

Gibt mehrere Elemente aus einer Liste zurück, nachdem sie entfernt wurden. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- LPOP

Gibt die ersten Elemente in einer Liste zurück, nachdem sie entfernt wurde. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- LPOS

Gibt den Index übereinstimmender Elemente in einer Liste zurück.

[Weitere Informationen](#)

- LPUSH

Stellt einer Liste ein oder mehrere Elemente voran. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- LPUSHX

Stellt einer Liste nur dann ein oder mehrere Elemente voran, wenn die Liste existiert.

[Weitere Informationen](#)

- LRANGE

Gibt einen Bereich von Elementen aus einer Liste zurück.

[Weitere Informationen](#)

- LREM

Entfernt Elemente aus einer Liste. Löscht die Liste, wenn das letzte Element entfernt wurde.

[Weitere Informationen](#)

- LSET

Legt den Wert eines Elements in einer Liste anhand seines Index fest.

[Weitere Informationen](#)

- LTRIM

Entfernt Elemente am Anfang und Ende einer Liste. Löscht die Liste, wenn alle Elemente gekürzt wurden.

[Weitere Informationen](#)

- RPOP

Gibt die letzten Elemente einer Liste zurück und entfernt sie. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- RPOPLPUSH

Gibt das letzte Element einer Liste zurück, nachdem es entfernt und in eine andere Liste verschoben wurde. Löscht die Liste, wenn das letzte Element gelöscht wurde.

[Weitere Informationen](#)

- RPUSH

Fügt ein oder mehrere Elemente an eine Liste an. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- RPUSHX

Fügt ein Element nur dann an eine Liste an, wenn die Liste existiert.

[Weitere Informationen](#)

## Pub/Sub-Befehle

 Note

PUBSUBBefehle verwenden intern ShardedPUBSUB, sodass Kanalnamen gemischt werden.

- PUBLISH

Sendet eine Nachricht an einen Kanal.

[Weitere Informationen](#)

- PUBSUB CHANNELS

Gibt die aktiven Kanäle zurück.

[Weitere Informationen](#)

- PUBSUB NUMSUB

Gibt die Anzahl der Subscriber von Kanälen zurück.

[Weitere Informationen](#)

- PUBSUB SHARDCHANNELS

Gibt die aktiven Shard-Kanäle zurück.

[PUBSUB-SHARDCHANNELS](#)

- PUBSUB SHARDNUMSUB

Gibt die Anzahl der Subscriber von Shard-Kanälen zurück.

[PUBSUB-SHARDNUMSUB](#)

- SPUBLISH

Sendet eine Nachricht an einen Shard-Kanal.

[Weitere Informationen](#)

- SSUBSCRIBE

Lauscht nach Nachrichten, die auf Shard-Kanälen veröffentlicht wurden.

[Weitere Informationen](#)

- SUBSCRIBE

Lauscht nach Nachrichten, die auf Kanälen veröffentlicht wurden.

[Weitere Informationen](#)

- SUNSUBSCRIBE

Beendet das Lauschen nach Nachrichten, die an Shard-Kanäle gesendet wurden.

[Weitere Informationen](#)

- UNSUBSCRIBE

Beendet das Lauschen nach Nachrichten, die an Kanäle gesendet wurden.

[Weitere Informationen](#)

## Befehle zur Skripterstellung

- EVAL

Führt ein serverseitiges Lua-Skript aus.

[Weitere Informationen](#)

- EVAL\_RO

Führt ein schreibgeschütztes serverseitiges Lua-Skript aus.

[Weitere Informationen](#)

- EVALSHA

Führt ein serverseitiges Lua-Skript per Digest aus. SHA1

[Weitere Informationen](#)

- EVALSHA\_RO

Führt ein schreibgeschütztes serverseitiges Lua-Skript per Digest aus. SHA1

[Weitere Informationen](#)

- SCRIPT EXISTS

Ermittelt, ob serverseitige Lua-Skripte im Skriptcache vorhanden sind.

#### [Weitere Informationen](#)

- SCRIPT FLUSH

Derzeit wird ein No-Op-Skript-Cache vom Service verwaltet.

#### [Weitere Informationen](#)

- SCRIPT LOAD

Lädt ein serverseitiges Lua-Skript in den Skript-Cache.

#### [Weitere Informationen](#)

### Befehle zur Serververwaltung

- ACL CAT

Listet die ACL Kategorien oder die Befehle innerhalb einer Kategorie auf.

#### [Weitere Informationen](#)

- ACL GENPASS

Generiert ein pseudozufälliges, sicheres Passwort, das zur Identifizierung von Benutzern verwendet werden kann. ACL

#### [Weitere Informationen](#)

- ACL GETUSER

Listet die ACL Regeln eines Benutzers auf.

#### [Weitere Informationen](#)

- ACL LIST

Gibt die effektiven Regeln im ACL Dateiformat aus.

#### [Weitere Informationen](#)

- ACL USERS

Listet alle ACL Benutzer auf.

### [Weitere Informationen](#)

- ACL WHOAMI

Gibt den authentifizierten Benutzernamen der aktuellen Verbindung zurück.

### [Weitere Informationen](#)

- DBSIZE

Gibt die Anzahl der Schlüssel in der aktuell ausgewählten Datenbank zurück. Es kann nicht garantiert werden, dass diese Operation in allen Slots atomar abläuft.

### [Weitere Informationen](#)

- COMMAND

Gibt detaillierte Informationen zu allen Befehlen zurück.

### [Weitere Informationen](#)

- COMMAND COUNT

Gibt eine Anzahl von Befehlen zurück.

### [Weitere Informationen](#)

- COMMAND DOCS

Gibt durch Dokumente belegte Informationen zu einem, mehreren oder allen Befehlen zurück.

### [Weitere Informationen](#)

- COMMAND GETKEYS

Extrahiert die Schlüsselnamen aus einem willkürlichen Befehl.

### [Weitere Informationen](#)

- COMMAND GETKEYSANDFLAGS

Extrahiert die Schlüsselnamen und Zugriffs-Flags für einen willkürlichen Befehl.

### [Weitere Informationen](#)

- **COMMAND INFO**

Gibt Informationen zu einem, mehreren oder allen Befehlen zurück.

[Weitere Informationen](#)

- **COMMAND LIST**

Gibt eine Liste von Befehlsnamen zurück.

[Weitere Informationen](#)

- **FLUSHALL**

Entfernt alle Schlüssel aus allen Datenbanken. Es kann nicht garantiert werden, dass diese Operation in allen Slots atomar abläuft.

[Weitere Informationen](#)

- **FLUSHDB**

Entfernt alle Schlüssel aus der aktuellen Datenbank. Es kann nicht garantiert werden, dass diese Operation in allen Slots atomar abläuft.

[Weitere Informationen](#)

- **INFO**

Gibt Informationen und Statistiken über den Server zurück.

[Weitere Informationen](#)

- **LOLWUT**

Zeigt Computergrafiken und die Valkey- oder OSS Redis-Version an.

[Weitere Informationen](#)

- **ROLE**

Gibt die Replikationsrolle zurück.

[Weitere Informationen](#)

- **TIME**

Gibt die Serverzeit zurück.

## [Weitere Informationen](#)

### Set-Befehle

- SADD

Fügt einem Set ein oder mehrere Mitglieder hinzu. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

## [Weitere Informationen](#)

- SCARDT

Gibt die Anzahl der Mitglieder in einem Set zurück.

## [Weitere Informationen](#)

- SDIFF

Gibt die Differenz zwischen mehreren Sets zurück.

## [Weitere Informationen](#)

- SDIFFSTORE

Speichert die Differenz zwischen mehreren Sets in einem Schlüssel.

## [Weitere Informationen](#)

- SINTER

Gibt den Schnittpunkt mehrerer Sets zurück.

## [Weitere Informationen](#)

- SINTERCARD

Gibt die Anzahl der Mitglieder des Schnittpunkts mehrerer Sets zurück.

## [Weitere Informationen](#)

- SINTERSTORE

Speichert den Schnittpunkt mehrerer Sets in einem Schlüssel.

[Weitere Informationen](#)

- SISMEMBER

Ermittelt, ob ein Mitglied zu einem Set gehört.

[Weitere Informationen](#)

- SMEMBERS

Gibt alle Mitglieder eines Sets zurück.

[Weitere Informationen](#)

- SMISMEMBER

Ermittelt, ob mehrere Mitglieder zu einem Set gehören.

[Weitere Informationen](#)

- SMOVE

Verschiebt ein Mitglied von einem Set in ein anderes.

[Weitere Informationen](#)

- SPOP

Gibt ein oder mehrere zufällige Mitglieder aus einer Gruppe zurück, nachdem sie entfernt wurden. Löscht das Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- SRANDMEMBER

Ruft ein oder mehrere zufällige Mitglieder aus einem Set ab.

[Weitere Informationen](#)

- SREM

Entfernt ein oder mehrere Mitglieder aus einem Set. Löscht das Set, wenn das letzte Mitglied entfernt wurde.

[Weitere Informationen](#)

- SSCAN

Iteriert über Mitglieder eines Sets.

### [Weitere Informationen](#)

- SUNION

Gibt die Vereinigung mehrerer Sets zurück.

### [Weitere Informationen](#)

- SUNIONSTORE

Speichert die Vereinigung mehrerer Sets in einem Schlüssel.

### [Weitere Informationen](#)

## Befehle zu Sorted Sets

- BZMPOP

Entfernt ein Mitglied aus einem oder mehreren sortierten Sets und gibt es nach Score zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht das sortierte Set, wenn das letzte Element gelöscht wurde.

### [Weitere Informationen](#)

- BZPOPMAX

Entfernt das Element mit dem höchsten Score aus einem oder mehreren sortierten Sets und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht das sortierte Set, wenn das letzte Element gelöscht wurde.

### [Weitere Informationen](#)

- BZPOPMIN

Entfernt das Element mit dem niedrigsten Score aus einem oder mehreren sortierten Sets und gibt es zurück. Blockiert, bis ein Element anderweitig verfügbar ist. Löscht das sortierte Set, wenn das letzte Element gelöscht wurde.

### [Weitere Informationen](#)

- ZADD

Fügt einem sortierten Set ein oder mehrere Mitglieder hinzu oder aktualisiert deren Scores. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

### [Weitere Informationen](#)

- ZCARD

Gibt die Anzahl der Mitglieder in einem sortierten Set zurück.

### [Weitere Informationen](#)

- ZCOUNT

Gibt die Anzahl der Mitglieder in einem sortierten Set zurück, deren Scores innerhalb eines bestimmten Bereichs liegen.

### [Weitere Informationen](#)

- ZDIFF

Gibt den Unterschied zwischen mehreren sortierten Sets zurück.

### [Weitere Informationen](#)

- ZDIFFSTORE

Speichert den Unterschied mehrerer sortierter Sets in einem Schlüssel.

### [Weitere Informationen](#)

- ZINCRBY

Inkrementiert den Score eines Mitglieds in einem sortierten Set.

### [Weitere Informationen](#)

- ZINTER

Gibt den Schnittpunkt mehrerer sortierter Sets zurück.

### [Weitere Informationen](#)

- ZINTERCARD

Gibt die Anzahl der Mitglieder des Schnittpunkts mehrerer sortierter Sets zurück.

[Weitere Informationen](#)

- ZINTERSTORE

Speichert den Schnittpunkt mehrerer sortierter Sets in einem Schlüssel.

[Weitere Informationen](#)

- ZLEXCOUNT

Gibt die Anzahl der Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs zurück.

[Weitere Informationen](#)

- ZMPOP

Gibt die Elemente mit dem höchsten oder niedrigsten Score aus einem oder mehreren sortierten Sets zurück, nachdem sie entfernt wurden. Löscht das sortierte Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- ZMSCORE

Gibt den Score eines oder mehrerer Mitglieder eines sortierten Sets zurück.

[Weitere Informationen](#)

- ZPOPMAX

Gibt die Mitglieder mit dem höchsten Score aus einem sortierten Set zurück, nachdem sie entfernt wurden. Löscht das sortierte Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- ZPOPMIN

Gibt die Mitglieder mit dem niedrigsten Score aus einem sortierten Set zurück, nachdem sie entfernt wurden. Löscht das sortierte Set, wenn das letzte Mitglied gelöscht wurde.

[Weitere Informationen](#)

- ZRANDMEMBER

Gibt ein oder mehrere zufällige Mitglieder aus einem sortierten Set zurück.

[Weitere Informationen](#)

- ZRANGE

Gibt Elemente in einem sortierten Set innerhalb eines Indexbereichs zurück.

[Weitere Informationen](#)

- ZRANGEBYLEX

Gibt Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs zurück.

[Weitere Informationen](#)

- ZRANGEBYSCORE

Gibt Elemente in einem sortierten Set innerhalb eines Score-Bereichs zurück.

[Weitere Informationen](#)

- ZRANGESTORE

Speichert einen Bereich von Elementen aus einem sortierten Set in einem Schlüssel.

[Weitere Informationen](#)

- ZRANK

Gibt den Index eines Elements in einem sortierten Set zurück, sortiert nach aufsteigenden Scores.

[Weitere Informationen](#)

- ZREM

Entfernt ein oder mehrere Elemente aus einem sortierten Set. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

[Weitere Informationen](#)

- ZREMRANGEBYLEX

Entfernt Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

[Weitere Informationen](#)

- ZREMRANGEBYRANK

Entfernt Elemente in einem sortierten Set innerhalb eines Indexbereichs. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

#### [Weitere Informationen](#)

- ZREMRANGEBYSCORE

Entfernt Elemente aus einem sortierten Set innerhalb eines Score-Bereichs. Löscht das sortierte Set, wenn alle Elemente entfernt wurden.

#### [Weitere Informationen](#)

- ZREVRANGE

Gibt Elemente in einem sortierten Set innerhalb eines Indexbereichs in umgekehrter Reihenfolge zurück.

#### [Weitere Informationen](#)

- ZREVRANGEBYLEX

Gibt Elemente in einem sortierten Set innerhalb eines lexikografischen Bereichs in umgekehrter Reihenfolge zurück.

#### [Weitere Informationen](#)

- ZREVRANGEBYSCORE

Gibt Elemente in einem sortierten Set innerhalb eines Score-Bereichs in umgekehrter Reihenfolge zurück.

#### [Weitere Informationen](#)

- ZREVRANK

Gibt den Index eines Elements in einem sortierten Set zurück, sortiert nach absteigenden Scores.

#### [Weitere Informationen](#)

- ZSCAN

Iteriert über Elemente und Scores eines sortierten Sets.

#### [Weitere Informationen](#)

- ZSCORE

Gibt den Score eines Elements in einem sortierten Set zurück.

[Weitere Informationen](#)

- ZUNION

Gibt die Vereinigung mehrerer sortierter Sets zurück.

[Weitere Informationen](#)

- ZUNIONSTORE

Speichert die Vereinigung mehrerer sortierter Sets in einem Schlüssel.

[Weitere Informationen](#)

## Stream-Befehle

- XACK

Gibt die Anzahl der Nachrichten zurück, die vom Mitglied der Verbrauchergruppe eines Streams erfolgreich bestätigt wurden.

[Weitere Informationen](#)

- XADD

Hängt eine neue Nachricht an einen Stream an. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- XAUTOCLAIM

Ändert oder erwirbt den Besitz von Nachrichten in einer Verbrauchergruppe, als ob die Nachrichten als Mitglied der Verbrauchergruppe zugestellt würden.

[Weitere Informationen](#)

- XCLAIM

Ändert oder erwirbt den Besitz einer Nachricht in einer Verbrauchergruppe, als ob die Nachricht als Mitglied der Verbrauchergruppe zugestellt würde.

[Weitere Informationen](#)

- XDEL

Gibt die Anzahl der Nachrichten zurück, nachdem sie aus einem Stream entfernt wurden.

[Weitere Informationen](#)

- XGROUP CREATE

Erstellt eine Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP CREATECONSUMER

Erstellt einen Verbraucher in einer Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP DELCONSUMER

Löscht einen Verbraucher aus einer Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP DESTROY

Löscht eine Verbrauchergruppe.

[Weitere Informationen](#)

- XGROUP SETID

Legt die zuletzt zugestellte ID einer Verbrauchergruppe fest.

[Weitere Informationen](#)

- XINFO CONSUMERS

Gibt eine Liste der Verbraucher in einer Verbrauchergruppe zurück.

[Weitere Informationen](#)

- XINFO GROUPS

Gibt eine Liste der Verbrauchergruppen eines Streams zurück.

### [Weitere Informationen](#)

- XINFO STREAM

Gibt Informationen zu einem Stream zurück.

### [Weitere Informationen](#)

- XLEN

Gibt die Anzahl der Nachrichten in einem Stream zurück.

### [Weitere Informationen](#)

- XPENDING

Gibt die Informationen und Einträge aus der Liste der ausstehenden Einträge einer Stream-Verbrauchergruppe zurück.

### [Weitere Informationen](#)

- XRANGE

Gibt die Nachrichten aus einem Stream innerhalb eines Bereichs von zurück. IDs

### [Weitere Informationen](#)

- XREAD

Gibt Nachrichten aus mehreren Streams zurück, deren IDs Anzahl größer als die angeforderten ist. Blockiert, bis eine Nachricht anderweitig verfügbar ist.

### [Weitere Informationen](#)

- XREADGROUP

Gibt neue oder historische Nachrichten aus einem Stream für einen Verbraucher in einer Gruppe zurück. Blockiert, bis eine Nachricht anderweitig verfügbar ist.

### [Weitere Informationen](#)

- XREVRANGE

Gibt die Nachrichten aus einem Stream innerhalb eines Bereichs von IDs in umgekehrter Reihenfolge zurück.

[Weitere Informationen](#)

- XTRIM

Löscht Nachrichten am Anfang eines Streams.

[Weitere Informationen](#)

## Zeichenfolgenbefehle

- APPEND

Hängt eine Zeichenfolge an den Wert eines Schlüssels an. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

[Weitere Informationen](#)

- DECR

Dekrementiert den Ganzzahlwert eines Schlüssels um eins. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- DECRBY

Dekrementiert eine Zahl vom Ganzzahlwert eines Schlüssels. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

[Weitere Informationen](#)

- GET

Gibt den Zeichenfolgenwert eines Schlüssels zurück.

[Weitere Informationen](#)

- GETDEL

Gibt den Zeichenfolgenwert eines Schlüssels zurück, nachdem der Schlüssel gelöscht wurde.

[Weitere Informationen](#)

- GETEX

Gibt den Zeichenfolgenwert eines Schlüssels zurück, nachdem dessen Ablaufzeit festgelegt wurde.

### [Weitere Informationen](#)

- GETRANGE

Gibt eine Teilzeichenfolge der Zeichenfolge zurück, die in einem Schlüssel gespeichert ist.

### [Weitere Informationen](#)

- GETSET

Gibt den vorherigen Zeichenfolgenwert eines Schlüssels zurück, nachdem dieser auf einen neuen Wert festgelegt wurde.

### [Weitere Informationen](#)

- INCR

Inkrementiert den Ganzzahlwert eines Schlüssels um eins. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

### [Weitere Informationen](#)

- INCRBY

Inkrementiert den Ganzzahlwert eines Schlüssels um eine Zahl. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

### [Weitere Informationen](#)

- INCRBYFLOAT

Inkrementiert den Gleitkommawert eines Schlüssels um eine Zahl. Verwendet 0 als Anfangswert, wenn der Schlüssel nicht existiert.

### [Weitere Informationen](#)

- LCS

Findet die längste gemeinsame Teilzeichenfolge.

### [Weitere Informationen](#)

- MGET

Gibt atomar die Zeichenfolgenwerte eines oder mehrerer Schlüssel zurück.

### [Weitere Informationen](#)

- MSET

Erstellt oder ändert atomar die Zeichenfolgenwerte eines oder mehrerer Schlüssel.

### [Weitere Informationen](#)

- MSETNX

Ändert die Zeichenfolgenwerte eines oder mehrerer Schlüssel nur dann atomar, wenn alle Schlüssel nicht existieren.

### [Weitere Informationen](#)

- PSETEX

Legt sowohl den Zeichenfolgenwert als auch die Ablaufzeit eines Schlüssels in Millisekunden fest. Der Schlüssel wird erstellt, wenn er nicht existiert.

### [Weitere Informationen](#)

- SET

Legt den Zeichenfolgenwert eines Schlüssels fest, wobei sein Typ ignoriert wird. Der Schlüssel wird erstellt, wenn er nicht existiert.

### [Weitere Informationen](#)

- SETEX

Legt den Zeichenfolgenwert und die Ablaufzeit eines Schlüssels fest. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

### [Weitere Informationen](#)

- SETNX

Legt den Zeichenfolgenwert eines Schlüssels nur fest, wenn der Schlüssel nicht existiert.

### [Weitere Informationen](#)

- SETRANGE

Überschreibt einen Teil eines Zeichenfolgenwerts durch einen anderen um einen Offset. Erstellt den Schlüssel, sofern er nicht vorhanden ist.

### [Weitere Informationen](#)

- STRLEN

Gibt die Länge eines Zeichenfolgenwerts zurück.

### [Weitere Informationen](#)

- SUBSTR

Gibt eine Teilzeichenfolge aus einem Zeichenfolgenwert zurück.

### [Weitere Informationen](#)

## Transaktionsbefehle

- DISCARD

Verwirft eine Transaktion.

### [Weitere Informationen](#)

- EXEC

Führt alle Befehle in einer Transaktion aus.

### [Weitere Informationen](#)

- MULTI

Startet eine Transaktion.

### [Weitere Informationen](#)

## Eingeschränkte Valkey- und Redis-Befehle OSS

ElastiCache schränkt den Zugriff auf bestimmte Cache-Engine-spezifische Befehle ein, die erweiterte Rechte erfordern, um ein Managed Service-Erlebnis zu bieten. Für Caches, die auf Redis betrieben werden, sind folgende Befehle nicht verfügbar:

- `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`

Darüber hinaus sind die folgenden Befehle für Serverless-Caches nicht verfügbar:

- `acl log`

- `client caching`
- `client getredir`
- `client id`
- `client info`
- `client kill`
- `client list`
- `client no-evict`
- `client pause`
- `client tracking`
- `client trackinginfo`
- `client unblock`
- `client unpause`
- `cluster count-failure-reports`
- `fcall`
- `fcall_ro`
- `function`
- `function delete`
- `function dump`
- `function flush`
- `function help`
- `function kill`
- `function list`
- `function load`
- `function restore`
- `function stats`
- `keys`
- `lastsave`
- `latency`

- latency doctor
- latency graph
- latency help
- latency histogram
- latency history
- latency latest
- latency reset
- memory
- memory doctor
- memory help
- memory malloc-stats
- memory purge
- memory stats
- memory usage
- monitor
- move
- object
- object encoding
- object freq
- object help
- object idletime
- object refcount
- pfdebug
- pfselftest
- psubscribe
- pubsub numpat
- punsubscribe
- script kill
- slowlog

- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `unwatch`
- `wait`
- `watch`

## Unterstützte Memcached-Befehle

ElastiCache Serverless for Memcached unterstützt alle [Memcached-Befehle in Open Source Memcached 1.6 mit Ausnahme](#) der folgenden:

- Client-Verbindungen erfordern TLS, weshalb das Protokoll nicht unterstützt wird. UDP
- Das Binärprotokoll wird nicht unterstützt, da es in Memcached 1.6 offiziell [veraltet](#) ist.
- GET/GETS-Befehle sind auf 16 KB begrenzt, um einen möglichen DoS-Angriff auf den Server durch Abrufen einer großen Anzahl von Schlüsseln zu vermeiden.
- Ein verzögerter `flush_all`-Befehl wird mit `CLIENT_ERROR` zurückgewiesen.
- Befehle, die die Engine konfigurieren oder interne Informationen zum Engine-Status oder zu Protokollen preisgeben, werden nicht unterstützt. Dazu gehören:
  - Für STATS-Befehle werden nur `stats` und `stats reset` unterstützt. Für andere Varianten wird `ERROR` zurückgegeben.
  - `lru / lru_crawler`-Änderung der LRU und der LRU Crawler-Einstellungen
  - `watch` – beobachtet Memcached-Serverprotokolle
  - `verbosity` – konfiguriert die Protokollstufe des Servers
  - `me`- Der Befehl `meta debug (me)` wird nicht unterstützt

## OSSKonfiguration und Limits von Valkey und Redis

Die Valkey- und OSS Redis-Engines bieten jeweils eine Reihe von Konfigurationsparametern, von denen einige in ElastiCache (RedisOSS) und andere nicht modifizierbar sind, um eine stabile Leistung und Zuverlässigkeit zu gewährleisten.

## Serverless-Caches

Für serverlose Caches werden keine Parametergruppen verwendet und die gesamte Valkey- oder Redis-Konfiguration kann nicht geändert werden. OSS Die folgenden Valkey- oder Redis-Parameter sind vorhanden: OSS

| Name                                | Details                            | Beschreibung                                                                                                                                                                                                                                  |
|-------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| acl-pubsub-default                  | allchannels                        | Standard-Pubsub-Kanalberechtigungen für ACL Benutzer im Cache.                                                                                                                                                                                |
| client-output-buffer-limit          | normal 0 0 0<br>pubsub 32mb 8mb 60 | Für normale Clients besteht kein Pufferlimit. PUB/SUBClients werden getrennt, wenn sie den 32 MiB-Backlog oder den 8 MiB-Backlog für 60 Sekunden durchbrechen.                                                                                |
| client-query-buffer-limit           | 1 GiB                              | Die maximale Größe eines einzelnen Client-Abfragepuffers. Darüber hinaus können Clients keine Anfrage mit mehr als 4 000 Argumenten stellen.                                                                                                  |
| cluster-allow-pubsubshard-when-down | yes                                | Dadurch kann der Cache Pub/Sub-Datenverkehr bereitstellen, während der Cache teilweise ausgefallen ist.                                                                                                                                       |
| cluster-allow-reads-when-down       | yes                                | Dadurch kann der Cache Lesedatenverkehr bereitstellen, während der Cache teilweise ausgefallen ist.                                                                                                                                           |
| cluster-enabled                     | yes                                | Für alle Serverless-Caches ist der Clustermodus aktiviert, sodass sie ihre Daten transparent auf mehrere Backend-Shards partitionieren können. Alle Slots werden den Clients so angezeigt, als gehörten sie einem einzigen virtuellen Knoten. |

| Name                                       | Details                   | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>cluster-require-full-coverage</code> | no                        | Wenn der Keyspace teilweise ausgefallen ist (d. h. auf mindestens einen Hash-Slot kann nicht zugegriffen werden), akzeptiert der Cache weiterhin Abfragen für den Teil des Keyspace, der noch abgedeckt ist. Der gesamte Keyspace wird immer von einem einzigen virtuellen Knoten in <code>cluster slots</code> „abgedeckt“.                                                                                                                                                                                                                                                                     |
| <code>lua-time-limit</code>                | 5000                      | <p>Die maximale Ausführungszeit für ein Lua-Skript in Millisekunden, bevor Maßnahmen zum Stoppen des Skripts ergriffen werden. ElastiCache</p> <p>Wenn sie überschritten <code>lua-time-limit</code> wird, geben alle Valkey- oder OSS Redis-Befehle möglicherweise einen Fehler in der Form <code>_____ - zurück. BUSY</code> Da dieser Status viele wichtige Valkey- oder OSS Redis-Operationen stören kann, ElastiCache wird zunächst ein Befehl ausgegeben. <code>SCRIPTKILL</code> Wenn dies nicht erfolgreich ist, ElastiCache werden Valkey oder Redis zwangsweise neu gestartet. OSS</p> |
| <code>maxclients</code>                    | 65000                     | Die maximale Anzahl von Clients, die zu jedem beliebigen Zeitpunkt mit dem Cache verbunden sein können. Weitere Verbindungen können erfolgreich hergestellt werden oder auch nicht.                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>maxmemory-policy</code>              | <code>volatile-lru</code> | Elemente mit einem TTL Satz werden nach der Schätzung <code>least-recently-used</code> (LRU) gelöscht, wenn das Speicherlimit eines Caches erreicht ist.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>notify-keyspace-events</code>        | (eine leere Zeichenfolge) | Keyspace-Ereignisse werden in Serverless-Caches derzeit nicht unterstützt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

| Name               | Details                               | Beschreibung                                                                                                                                                                                                                                          |
|--------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| port               | Primärer Port: 6379<br>Leseport: 6380 | Serverless-Caches kündigen zwei Ports mit demselben Hostnamen an. Der primäre Port lässt Schreib- und Lesevorgänge zu, wohingegen der Leseport mithilfe des Befehls READONLY letztendlich konsistente Lesevorgänge mit niedrigerer Latenz ermöglicht. |
| proto-max-bulk-len | 512 MiB                               | Die maximale Größe einer einzelnen Elementanforderung.                                                                                                                                                                                                |
| timeout            | 0                                     | Zu einer bestimmten Leerlaufzeit werden die Verbindungen mit Clients nicht gewaltsam getrennt, sie können jedoch aus Gründen des Lastausgleichs im stabilen Zustand getrennt werden.                                                                  |

Darüber hinaus gelten die folgenden Limits:

| Name                      | Details | Beschreibung                                                                                                                                                                        |
|---------------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Länge des Schlüsselnamens | 4 KiB   | Die maximale Größe für einen einzelnen Valkey- oder Redis-Schlüssel- oder Kanalnamen OSS. Clients, die auf Schlüssel verweisen, die größer sind, wird eine Fehlermeldung angezeigt. |
| Größe des Lua-Skripts     | 4 MiB   | Die maximale Größe eines einzelnen Valkey- oder OSS Redis-Lua-Skripts. Versuche, ein Lua-Skript zu laden, das größer ist als dieses, führen zu einer Fehlermeldung.                 |
| Slot-Größe                | 32 GiB  | Die maximale Größe eines einzelnen Valkey- oder Redis-Hash-Slots. OSS Clients, die versuchen, mehr Daten auf einem einzelnen                                                        |

| Name | Details | Beschreibung                                                                                                                                                                                                  |
|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      |         | Valkey- oder OSS Redis-Steckplatz zu speichern, lösen die Räumungsrichtlinie für den Steckplatz aus, und wenn keine Schlüssel entfernt werden können, wird ein Fehler wegen Speichermangel ( ) angezeigt. 00M |

## Selbst entworfene Cluster

Informationen über Standardwerte der Konfigurationsparameter und die konfigurierbaren Werte für selbst entworfene Cluster finden Sie unter [Valkey- und Redis-Parameter OSS](#). Die Standardwerte werden generell empfohlen, es sei denn, Sie haben einen bestimmten Anwendungsfall, bei dem sie außer Kraft gesetzt werden müssen.

## IPv6Kundenbeispiele für Valkey, Redis OSS und Memcached

ElastiCache ist kompatibel mit Valkey, Redis OSS und Memcached. Das bedeutet, dass Clients, die IPv6 Verbindungen unterstützen, in der Lage sein sollten, eine Verbindung zu IPv6 aktivierten ElastiCache (Memcached-) Clustern herzustellen. Bei der Interaktion mit IPv6 aktivierten Ressourcen sind einige Vorbehalte zu beachten.

Empfehlungen zur Konfiguration von [Valkey- und Redis-Clients für Ressourcen finden Sie im Blogbeitrag Best Practices](#) für Valkey- und OSS Redis-Clients im AWS Datenbank-Blog. ElastiCache

Im Folgenden finden Sie bewährte Methoden für die Interaktion mit IPv6 aktivierten ElastiCache Ressourcen mithilfe häufig verwendeter Open-Source-Clientbibliotheken.

### Validierte Kunden mit Valkey und Redis OSS

ElastiCache ist kompatibel mit Valkey und Open-Source-Redis. OSS Das bedeutet, dass Valkey- und OSS Open-Source-Redis-Clients, die IPv6 Verbindungen unterstützen, in der Lage sein sollten, eine Verbindung zu IPv6 aktivierten ElastiCache (Redis-) Clustern herzustellen. OSS Darüber hinaus wurden mehrere der beliebtesten Python- und Java-Clients speziell getestet und validiert, sodass sie mit allen unterstützten Netzwerktypkonfigurationen (IPv4nur, IPv6 nur und Dual Stack) funktionieren.

Die folgenden Clients wurden speziell dafür validiert, dass sie mit allen unterstützten Netzwerktypkonfigurationen für Valkey und OSS Redis funktionieren.

Validierte Clients:

- [Redis Py \(\) – 4.1.2](#)
- [Kopfsalat — Version: 6.1.6. RELEASE](#)
- [Jedis – Version: 3.6.0](#)

## Best Practices für Kunden (Valkey und OSS Redis)

Lernen Sie bewährte Methoden für gängige Szenarien kennen und folgen Sie den Codebeispielen einiger der beliebtesten OSS Open-Source-Clientbibliotheken von Valkey und Redis (redis-py, PHPRedis, und Lettuce) sowie von Best Practices für die Interaktion mit ElastiCache Ressourcen mit häufig verwendeten Open-Source-Memcached-Clientbibliotheken.

### Themen

- [Große Anzahl von Verbindungen \(Valkey und OSS Redis\)](#)
- [Cluster-Client-Erkennung und exponentielles Backoff \(Valkey und Redis\) OSS](#)
- [Konfigurieren Sie ein clientseitiges Timeout \(Valkey und Redis\) OSS](#)
- [Konfigurieren Sie ein serverseitiges Leerlauf-Timeout \(Valkey und Redis\) OSS](#)
- [Lua-Skripte](#)
- [Speichern großer zusammengesetzter Artikel \(Valkey und OSS Redis\)](#)
- [Konfiguration des Salat-Clients \(Valkey und Redis\) OSS](#)
- [Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster \(Valkey und Redis\) OSS](#)

## Große Anzahl von Verbindungen (Valkey und OSS Redis)

Serverlose Caches und einzelne ElastiCache (Redis-OSS) Knoten unterstützen bis zu 65.000 gleichzeitige Client-Verbindungen. Zur Optimierung der Leistung empfehlen wir jedoch, dass Client-Anwendungen nicht ständig mit diesem Verbindungsaufkommen arbeiten. Valkey und Redis verfügen OSS jeweils über einen Single-Thread-Prozess, der auf einer Ereignisschleife basiert, in der eingehende Client-Anfragen sequentiell bearbeitet werden. Das bedeutet, dass die Antwortzeit eines bestimmten Clients mit zunehmender Anzahl verbundener Clients länger wird.

Sie können die folgenden Maßnahmen ergreifen, um einen Verbindungsengpass auf einem Valkey- oder Redis-Server zu vermeiden: OSS

- Führen Sie Lesevorgänge von Read Replicas aus durch. Dies kann durch die Verwendung der ElastiCache Leser-Endpunkte im deaktivierten Clustermodus oder durch die Verwendung von

Replikaten für Lesevorgänge im aktivierten Clustermodus, einschließlich eines serverlosen Caches, erreicht werden.

- Verteilen Sie den Schreibdatenverkehr auf mehrere Primärknoten. Es gibt zwei Methoden dafür: Sie können einen Valkey- oder Redis-Cluster mit mehreren Shards und einem Client verwenden, der den OSS Clustermodus unterstützt. Sie können auch im deaktivierten Cluster-Modus mit clientseitigem Sharding auf mehrere Primärknoten schreiben. Dies erfolgt automatisch in einem Serverless-Cache.
- Verwenden Sie einen Verbindungspool, sofern dieser in Ihrer Client-Bibliothek verfügbar ist.

Im Allgemeinen ist das Herstellen einer TCP Verbindung im Vergleich zu typischen Valkey- oder Redis-Befehlen ein rechenintensiver Vorgang. OSS Beispielsweise ist die Bearbeitung einer SET GET /-Anforderung um eine Größenordnung schneller, wenn eine bestehende Verbindung wiederverwendet wird. Durch die Verwendung eines Client-Verbindungspools mit begrenzter Größe wird der Aufwand für die Verbindungsverwaltung reduziert. Außerdem wird dadurch die Anzahl der gleichzeitig eingehenden Verbindungen von der Client-Anwendung begrenzt.

Das folgende Codebeispiel PHPRedis zeigt, dass für jede neue Benutzeranfrage eine neue Verbindung erstellt wird:

```
$redis = new Redis();
if ($redis->connect($HOST, $PORT) != TRUE) {
 //ERROR: connection failed
 return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Wir haben diesen Code in einer Schleife auf einer Amazon Elastic Compute Cloud (AmazonEC2) -Instance verglichen, die mit einem Graviton2-Knoten (m6g.2xlarge) (Redis) verbunden ist. ElastiCache OSS Client und Server wurden in derselben Availability Zone platziert. Die durchschnittliche Latenz des gesamten Vorgangs betrug 2,82 Millisekunden.

Beim Aktualisieren des Codes und bei Verwendung persistenter Verbindungen sowie eines Verbindungspools betrug die durchschnittliche Latenz des gesamten Vorgangs 0,21 Millisekunden:

```
$redis = new Redis();
if ($redis->pconnect($HOST, $PORT) != TRUE) {
```

```
// ERROR: connection failed
return;
}
$redis->set($key, $value);
unset($redis);
$redis = NULL;
```

Erforderliche Konfigurationen für redis.ini:

- `redis.pconnect.pooling_enabled=1`
- `redis.pconnect.connection_limit=10`

Der folgende Code ist ein Beispiel für einen [Redis-py-Verbindungspool](#):

```
conn = Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
conn.set(key, value)
```

Der folgende Code ist ein Beispiel für einen [Lettuce-Verbindungspool](#):

```
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);
}
```

## Cluster-Client-Erkennung und exponentielles Backoff (Valkey und Redis) OSS

Wenn Sie im aktivierten Clustermodus eine Verbindung zu einem ElastiCache Valkey- oder OSS Redis-Cluster herstellen, muss die entsprechende Clientbibliothek clusterfähig sein. Die Clients müssen eine Zuordnung der Hash-Slots zu den entsprechenden Knoten im Cluster abrufen, um Anfragen an die richtigen Knoten zu senden und den zusätzlichen Leistungsaufwand bei der Bearbeitung von Cluster-Umleitungen zu vermeiden. Aus diesem Grund muss der Client in zwei verschiedenen Situationen eine vollständige Liste der Slots und der zugewiesenen Knoten erkennen:

- Der Client ist initialisiert und muss die anfängliche Slot-Konfiguration auffüllen.

- Eine MOVED Umleitung wird vom Server empfangen, z. B. bei einem Failover, bei dem alle vom früheren Primärknoten bedienten Steckplätze vom Replikant übernommen werden, oder bei einem Re-Sharding, wenn Steckplätze vom primären Quellknoten zum primären Zielknoten verschoben werden

Die Client-Erkennung erfolgt in der Regel durch Ausgabe eines CLUSTER NODE Befehls CLUSTER SLOT oder an den Valkey- oder Redis-Server. OSS Wir empfehlen CLUSTER SLOT diese Methode, da sie den Satz von Steckplatzbereichen und die zugehörigen Primär- und Replikatknoten an den Client zurückgibt. Dies erfordert kein zusätzliches Parsen durch den Client und ist effizienter.

Je nach Clustertopologie kann die Größe der Antwort auf den CLUSTER SLOT Befehl je nach Clustergröße variieren. Größere Cluster mit mehr Knoten erzeugen eine größere Antwort. Daher muss sichergestellt werden, dass die Anzahl der Clients, die die Cluster-Topologie ermitteln, nicht unbegrenzt zunimmt. Wenn beispielsweise die Client-Anwendung startet oder die Verbindung mit dem Server unterbrochen wird und die Cluster-Erkennung durchgeführt werden muss, besteht ein häufiger Fehler darin, dass die Client-Anwendung mehrere Wiederverbindungs- und Erkennungsanforderungen auslöst, ohne dass bei einem erneuten Versuch ein exponentielles Backoff erfolgt. Dies kann dazu führen, dass der Valkey- oder OSS Redis-Server über einen längeren Zeitraum nicht reagiert und die CPU Auslastung bei 100% liegt. Der Ausfall verlängert sich, wenn jeder CLUSTER SLOT Befehl eine große Anzahl von Knoten im Clusterbus verarbeiten muss. Wir haben in der Vergangenheit mehrere Client-Ausfälle aufgrund dieses Verhaltens in einer Reihe verschiedener Sprachen beobachtet, darunter Python (redis-py-cluster) und Java (Lettuce und Redisson).

In einem Serverless-Cache werden viele der Probleme automatisch behoben, da die angekündigte Cluster-Topologie statisch ist und aus zwei Einträgen besteht: einem Schreib- und einem Lese-Endpunkt. Die Cluster-Erkennung wird außerdem automatisch auf mehrere Knoten verteilt, wenn der Cache-Endpunkt verwendet wird. Die folgenden Empfehlungen sind jedoch nach wie vor nützlich.

Wir empfehlen Folgendes, um die Auswirkungen eines plötzlichen Zustroms von Verbindungs- und Erkennungsanforderungen zu minimieren:

- Implementieren Sie einen Client-Verbindungspool mit einer begrenzten Größe, um die Anzahl der gleichzeitig eingehenden Verbindungen von der Client-Anwendung zu begrenzen.
- Wenn der Client aufgrund eines Timeouts die Verbindung mit dem Server unterbricht, versuchen Sie es erneut mit exponentiellem Backoff mit Jitter. Dadurch wird vermieden, dass mehrere Clients den Server gleichzeitig überlasten.

- Verwenden Sie die Anleitung unter [Verbindungsendpunkte finden in ElastiCache](#), um den Cluster-Endpoint zur Durchführung der Cluster-Erkennung zu finden. Dadurch verteilen Sie die Erkennungslast auf alle Knoten im Cluster (bis zu 90), anstatt einige fest codierte Seed-Knoten im Cluster zu erhalten.

Im Folgenden finden Sie einige Codebeispiele für die exponentielle Backoff-Wiederholungslogik in Redis-py, und Lettuce. PHPRedis

Beispiel 1 für Backoff-Logik: redis-py

Redis-py verfügt über einen integrierten Wiederholungsmechanismus, der unmittelbar nach einem Fehler einen erneuten Versuch vornimmt. [Dieser Mechanismus kann durch das Argument aktiviert werden, das beim Erstellen eines Redis-Objekts angegeben wird](#) `retry_on_timeout`. OSS Hier demonstrieren wir einen benutzerdefinierten Wiederholungsmechanismus mit exponentiellem Backoff und Jitter. Wir haben eine Pull-Anfrage gesendet, um exponentielles Backoff in [redis-py \(#1494\)](#) nativ zu implementieren. Künftig ist eine manuelle Implementierung möglicherweise nicht mehr erforderlich.

```
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
```

Sie können dann den folgenden Code verwenden, um einen Wert festzulegen:

```
client = redis.Redis(connection_pool=redis.BlockingConnectionPool(host=HOST,
 max_connections=10))
res = run_with_backoff(lambda: client.set("key", "value"))
print(res)
```

Abhängig von Ihrem Workload möchten Sie möglicherweise den Basis-Backoff-Wert von 1 Sekunde auf einige Dutzende oder Hunderte von Millisekunden für latenzempfindliche Workloads ändern.

### Beispiel 2 für Backoff-Logik: PHPRedis

PHPRedis verfügt über einen integrierten Wiederholungsmechanismus, der (nicht konfigurierbar) maximal 10 Wiederholungen vornimmt. Es gibt eine konfigurierbare Verzögerung zwischen den Wiederholungsversuchen (mit einem Jitter ab dem zweiten Versuch). Weitere Informationen finden Sie im folgenden [Codebeispiel](#). [Wir haben eine Pull-Anfrage zur nativen Implementierung des exponentiellen Backoffs in PHPRedis\(#1986\) eingereicht, die inzwischen zusammengeführt und dokumentiert wurde](#). Für diejenigen, die die neueste Version von verwenden PHPRedis, ist eine manuelle Implementierung nicht erforderlich, aber wir haben hier die Referenz für die Versionen früherer Versionen aufgenommen. Im Folgenden finden Sie zunächst ein Codebeispiel, das die Verzögerung des Wiederholungsmechanismus konfiguriert:

```
$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);
```

### Beispiel 3 für Backoff-Logik: Lettuce

Lettuce verfügt über integrierte Wiederholungsmechanismen, die auf den exponentiellen Backoff-Strategien basieren, wie im Beitrag [Exponentielles Backoff und Jitter](#) beschrieben. Im Folgenden finden Sie einen Codeauszug, der den vollständigen Jitter-Ansatz zeigt:

```
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();
}

}
}
```

## Konfigurieren Sie ein clientseitiges Timeout (Valkey und Redis) OSS

### Konfiguration des clientseitigen Timeouts

Konfigurieren Sie das clientseitige Timeout so, dass der Server genügend Zeit hat, die Anfrage zu verarbeiten und die Antwort zu generieren. Dies ermöglicht einen Fail-Fast, wenn die Verbindung zum Server nicht hergestellt werden kann. Bestimmte Valkey- oder OSS Redis-Befehle können rechenintensiver sein als andere. Zum Beispiel Lua-Skripte oder MULTI EXEC /-Transaktionen, die mehrere Befehle enthalten, die atomar ausgeführt werden müssen. Im Allgemeinen wird ein höheres clientseitiges Timeout empfohlen, um ein Timeout des Clients zu vermeiden, bevor die Antwort vom Server empfangen wird. Dies umfasst:

- Ausführen von Befehlen über mehrere Tasten
- Ausführen von MULTI EXEC /Transactions oder Lua-Skripten, die aus mehreren einzelnen Valkey- oder Redis-Befehlen bestehen OSS

- Lesen von großen Werten
- Durchführen von Blockierungsvorgängen wie BLPOP

Im Falle eines Blockierungsvorgangs wie ist es am besten BLPOP, das Befehls-Timeout auf einen Wert zu setzen, der unter dem Socket-Timeout liegt.

Im Folgenden finden Sie Codebeispiele für die Implementierung eines clientseitigen Timeouts in redis-py, und Lettuce. PHPRedis

Beispiel 1 für eine Timeout-Konfiguration: redis-py

Im Folgenden finden Sie ein Codebeispiel für 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)
```

Beispiel 2 für eine Timeout-Konfiguration: PHPRedis

Das Folgende ist ein Codebeispiel mit 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
```

### Beispiel 3 für eine Timeout-Konfiguration: Lettuce

Im Folgenden finden Sie ein Codebeispiel für Lettuce:

```
// connect to Redis server and give every command a 2 second timeout
public static void main(String[] args)
{
 RedisClient client = null;
 StatefulRedisConnection<String, String> connection = null;
 try {
 client = RedisClient.create(RedisURI.create(HOST, PORT));
 client.setOptions(ClientOptions.builder()
 .socketOptions(SocketOptions.builder().connectTimeout(Duration.ofMillis(100)).build()) //
 100 millisecond connection timeout
 .timeoutOptions(TimeoutOptions.builder().fixedTimeout(Duration.ofSeconds(2)).build()) //
 2 second command timeout
 .build());

 // use the connection pool from above example

 commands.set("key", "value"); // will timeout after 2 seconds
 commands.blpop(1, "list"); // BLPPOP with 1 second timeout
 } finally {
 if (connection != null) {
 connection.close();
 }

 if (client != null){
 client.shutdown();
 }
 }
}
```

## Konfigurieren Sie ein serverseitiges Leerlauf-Timeout (Valkey und Redis) OSS

Wir haben Fälle beobachtet, in denen mit der Anwendung eines Kunden eine hohe Anzahl inaktiver Clients verbunden war, diese aber keine aktiven Befehle sendet. In solchen Szenarien können Sie alle 65 000 Verbindungen mit einer hohen Anzahl inaktiver Clients ausschöpfen. Zur Vermeidung solcher Szenarien konfigurieren Sie die Timeout-Einstellung auf dem Server entsprechend über

[Valkey- und Redis-Parameter OSS](#). Dadurch wird sichergestellt, dass der Server die Verbindung inaktiver Clients aktiv trennt, um eine Erhöhung der Anzahl von Verbindungen zu vermeiden. Diese Einstellung ist für Serverless-Caches nicht verfügbar.

## Lua-Skripte

Valkey und Redis OSS unterstützen mehr als 200 Befehle, einschließlich solcher zum Ausführen von Lua-Skripten. Bei Lua-Skripten gibt es jedoch mehrere Fallstricke, die sich auf den Speicher und die Verfügbarkeit von Valkey oder Redis auswirken können. OSS

### Nicht parametrisierte Lua-Skripte

Jedes Lua-Skript wird auf dem Valkey- oder Redis-Server zwischengespeichert, bevor es ausgeführt wird. OSS Unparametrisierte Lua-Skripten sind einzigartig, was dazu führen kann, dass der Valkey- oder OSS Redis-Server eine große Anzahl von Lua-Skripten speichert und mehr Speicher verbraucht. Um dem entgegenzuwirken, stellen Sie sicher, dass alle Lua-Skripten parametrisiert sind und bei Bedarf regelmäßig zwischengespeicherte Lua-Skripts bereinigen. `SCRIPT FLUSH`

Das folgende Codebeispiel zeigt, wie parametrisierte Skripte verwendet werden. Zunächst haben wir ein Beispiel für einen nicht parametrisierten Ansatz, der zu drei verschiedenen zwischengespeicherten Lua-Skripten führt. Dieser Ansatz wird nicht empfohlen:

```
eval "return redis.call('set','key1','1')" 0
eval "return redis.call('set','key2','2')" 0
eval "return redis.call('set','key3','3')" 0
```

Verwenden Sie stattdessen das folgende Muster, um ein einzelnes Skript zu erstellen, das übergebene Parameter akzeptieren kann:

```
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
```

### Lang andauernde Lua-Skripte

Lua-Skripte können mehrere Befehle atomar ausführen, sodass die Ausführung länger dauern kann als bei einem normalen Valkey- oder Redis-Befehl. OSS Wenn das Lua-Skript nur schreibgeschützte Operationen ausführt, können Sie es zwischendurch beenden. Sobald das Lua-Skript jedoch einen Schreibvorgang ausführt, kann es nicht mehr beendet werden und muss vollständig ausgeführt werden. Ein mutierendes Lua-Skript mit langer Laufzeit kann dazu führen, dass der Valkey- oder

OSS Redis-Server lange Zeit nicht reagiert. Zur Behebung dieses Problems vermeiden Sie lang andauernde Lua-Skripte und testen Sie das Skript in einer Vorproduktionsumgebung.

### Lua-Skript mit Stealth-Schreibvorgängen

Es gibt mehrere Möglichkeiten, wie ein Lua-Skript weiterhin neue Daten in Valkey oder Redis schreiben kann, auch wenn Valkey oder OSS Redis beendet ist: `OSS maxmemory`

- Das Skript startet, wenn sich der Valkey- oder OSS Redis-Server unten `maxmemory` befindet, und enthält mehrere Schreiboperationen
- Der erste Schreibbefehl des Skripts verbraucht keinen Speicher (z. B. `DEL`), gefolgt von weiteren Schreibvorgängen, die Speicher verbrauchen
- Sie können dieses Problem verringern, indem Sie auf einem anderen Valkey- oder OSS Redis-Server als eine geeignete Räumungsrichtlinie konfigurieren. `noeviction` Auf diese Weise kann Redis Elemente OSS entfernen und Speicherplatz zwischen Lua-Skripten freigeben.

### Speichern großer zusammengesetzter Artikel (Valkey und OSS Redis)

In einigen Szenarien kann eine Anwendung große zusammengesetzte Elemente in Valkey oder Redis speichern OSS (z. B. einen Hash-Datensatz mit mehreren GB). Diese Vorgehensweise wird nicht empfohlen, da sie häufig zu Leistungsproblemen in Valkey oder Redis führt. OSS Der Client kann beispielsweise einen `HGETALL` Befehl ausführen, um die gesamte Multi-GB-Hashsammlung abzurufen. Dies kann zu erheblichem Speicherdruck auf den Valkey- oder OSS Redis-Server führen, der das große Element im Client-Ausgabepuffer zwischenspeichert. Außerdem werden bei der Steckplatzmigration im Clustermodus ElastiCache keine Steckplätze migriert, die Elemente mit einer serialisierten Größe von mehr als 256 MB enthalten.

Zur Lösung der Probleme mit großen Elementen wird Folgendes empfohlen:

- Teilen Sie das große zusammengesetzte Element in mehrere kleinere Objekte auf. Teilen Sie beispielsweise eine große Hash-Sammlung in einzelne Schlüssel-Wert-Felder auf, wobei das Schlüsselnamensschema die Sammlung angemessen widerspiegelt, z. B. indem Sie ein gemeinsames Präfix im Schlüsselnamen verwenden, um die Sammlung von Elementen zu identifizieren. Wenn Sie atomar auf mehrere Felder in derselben Sammlung zugreifen müssen, können Sie den `MGET` Befehl verwenden, um mehrere Schlüsselwerte in demselben Befehl abzurufen.
- Wenn Sie alle Optionen geprüft haben und den großen Sammlungsdatensatz immer noch nicht aufteilen können, versuchen Sie, Befehle zu verwenden, die auf eine Teilmenge der Daten in

der Sammlung statt auf die gesamte Sammlung angewendet werden. Vermeiden Sie einen Anwendungsfall, bei dem Sie die gesamte Sammlung mit mehreren GB mit demselben Befehl atomar abrufen müssen. Ein Beispiel ist die Verwendung von HGET HMGET Or-Befehlen anstelle von HGETALL On-Hash-Sammlungen.

## Konfiguration des Salat-Clients (Valkey und Redis) OSS

In diesem Abschnitt werden die empfohlenen Konfigurationsoptionen für Java und Lettuce sowie deren Anwendung auf Cluster beschrieben. ElastiCache

Die Empfehlungen in diesem Abschnitt wurden mit Lettuce Version 6.2.2 getestet.

### Themen

- [Beispiel: Lettuce-Konfiguration für den Clustermodus, aktiviert TLS](#)
- [Beispiel: Die Lettuce-Konfiguration für den Cluster-Modus ist deaktiviert, aktiviert TLS](#)

### Java-Cache DNS TTL

Die virtuelle Java-Maschine (JVM) speichert DNS Namenssuchvorgänge im Cache. Wenn der einen Hostnamen in eine IP-Adresse JVM auflöst, speichert er die IP-Adresse für einen bestimmten Zeitraum im Cache, der als ( ) bezeichnet wird. time-to-liveTTL

Die Wahl des TTL Werts ist ein Kompromiss zwischen Latenz und Reaktionsfähigkeit auf Veränderungen. Je kürzerTTLs, desto schneller erkennen DNS Resolver Aktualisierungen im Cluster. DNS Dadurch kann Ihre Anwendung schneller auf Ersetzungen oder andere Workflows reagieren, denen Ihr Cluster unterzogen wird. Wenn der Wert jedoch zu niedrig TTL ist, erhöht sich das Abfragevolumen, was die Latenz Ihrer Anwendung erhöhen kann. Es gibt zwar keinen korrekten TTL Wert, aber es lohnt sich, bei der Festlegung Ihres TTL Werts zu berücksichtigen, wie lange Sie es sich leisten können, auf das Wirksamwerden einer Änderung zu warten.

Da ElastiCache Knoten DNS Namenseinträge verwenden, die sich ändern können, empfehlen wir Ihnen, Ihren JVM mit einem Tiefstwert TTL von 5 bis 10 Sekunden zu konfigurieren. Dadurch wird sichergestellt, dass Ihre Anwendung, wenn sich die IP-Adresse eines Knotens ändert, die neue IP-Adresse der Ressource empfangen und verwenden kann, indem sie den DNS Eintrag erneut abfragt.

Bei einigen Java-Konfigurationen TTL ist die JVM Standardeinstellung so eingestellt, dass DNS Einträge erst aktualisiert werden, wenn der neu gestartet JVM wird.

Einzelheiten zum Einstellen Ihres JVM TTL finden Sie unter [So legen Sie den JVM TTL fest.](#)

## Lettuce-Version

Wir empfehlen die Verwendung der Lettuce-Version 6.2.2 oder höher.

## Endpunkte

Wenn Sie Cluster verwenden, für die der Cluster-Modus aktiviert ist, legen Sie den `redisUri` auf den Endpunkt der Cluster-Konfiguration fest. Die DNS Suche danach URI gibt eine Liste aller verfügbaren Knoten im Cluster zurück und wird während der Cluster-Initialisierung nach dem Zufallsprinzip in einen von ihnen aufgelöst. Weitere Informationen zur Funktionsweise der Topologieaktualisierung finden Sie weiter unten in `dynamicRefreshResources` diesem Thema.

## SocketOption

Aktivieren. [KeepAlive](#) Durch die Aktivierung dieser Option wird die Notwendigkeit verringert, während der Befehlslaufzeit ausgefallene Verbindungen zu behandeln.

Stellen Sie sicher, dass Sie das [Verbindungs-Timeout](#) entsprechend Ihren Anwendungsanforderungen und Ihrer Workload festlegen. Weitere Informationen finden Sie im Abschnitt „Timeouts“ weiter unten in diesem Thema.

ClusterClientOption: Client-Optionen mit aktiviertem Clustermodus

Aktiviert [AutoReconnect](#), wenn die Verbindung unterbrochen wird.

Eingestellt [CommandTimeout](#). Weitere Einzelheiten finden Sie im Abschnitt „Timeouts“ weiter unten in diesem Thema.

Legt fest [nodeFilter](#), dass ausgefallene Knoten aus der Topologie herausgefiltert werden. Lettuce speichert alle Knoten, die in der Ausgabe „Clusterknoten“ gefunden werden (einschließlich Knoten mit dem FAIL Status `PFAIL/`), in den „Partitionen“ des Clients (auch bekannt als Shards). Während der Erstellung der Cluster-Topologie wird versucht, eine Verbindung mit allen Knoten der Partition herzustellen. Dieses Verhalten von Lettuce, bei dem ausgefallene Knoten hinzugefügt werden, kann Verbindungsfehlern (oder Warnungen) verursachen, wenn Knoten aus einem beliebigen Grund ersetzt werden.

Wenn beispielsweise ein Failover abgeschlossen ist und der Cluster den Wiederherstellungsprozess startet, während der aktualisiert `clusterTopology` wird, gibt es in der Clusterbus-Knotenübersicht einen kurzen Zeitraum, in dem der ausgefallene Knoten als Knoten aufgeführt wird, bevor er vollständig aus der FAIL Topologie entfernt wird. Während dieser Zeit betrachtet der Lettuce-Client ihn als

fehlerfreien Knoten und stellt kontinuierlich eine Verbindung zu ihm her. Dies führt zu einem Fehler, nachdem alle Wiederholungsversuche durchgeführt wurden.

Beispielsweise:

```
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

Die Knotenfilterung wird am besten verwendet, wenn sie auf true `DynamicRefreshSources` gesetzt ist. Andernfalls wird der Primärknoten eines Shards herausgefiltert, wenn die Topologieansicht von einem einzelnen problematischen Seed-Knoten übernommen wird, der diesen primären Knoten als ausgefallen ansieht. Dies führt dazu, dass die Slots nicht abgedeckt werden. Wenn mehrere Seed-Knoten vorhanden sind (wenn wahr), verringert `DynamicRefreshSources` sich die Wahrscheinlichkeit, dass dieses Problem auftritt, da zumindest einige der Seed-Knoten nach einem Failover mit dem neu hochgestuften Primärknoten über eine aktualisierte Topologieansicht verfügen sollten.

`ClusterTopologyRefreshOptions`: Optionen zur Steuerung der Aktualisierung der Clustertopologie auf dem Client mit aktiviertem Clustermodus

#### Note

Cluster mit deaktiviertem Cluster-Modus unterstützen die Befehle zur Cluster-Ermittlung nicht und sind nicht mit allen Funktionen zur dynamischen Topologieermittlung des Clients kompatibel.

Der Clustermodus ist deaktiviert mit ElastiCache ist nicht mit dem von Lettuce kompatibel. `MasterSlaveTopologyRefresh` Stattdessen können Sie für einen deaktivierten Cluster-

Modus einen `StaticMasterReplicaTopologyProvider` konfigurieren und die Lese- und Schreibendpunkte des Clusters bereitstellen.

Weitere Informationen zum Herstellen einer Verbindung mit Clustern mit deaktiviertem Cluster-Modus finden Sie unter [Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Wenn Sie die Funktion zur dynamischen Topologieermittlung von Lettuce verwenden möchten, können Sie einen Cluster mit aktiviertem Cluster-Modus mit derselben Shard-Konfiguration wie Ihr vorhandener Cluster erstellen. Für Cluster mit aktiviertem Cluster-Modus empfehlen wir jedoch, mindestens 3 Shards mit mindestens einem Replikat zu konfigurieren, um schnelles Failover zu unterstützen.

Aktiviert. [enablePeriodicRefresh](#) Dadurch werden regelmäßige Updates der Clustertopologie aktiviert, sodass der Client die Clustertopologie in Intervallen von `refreshPeriod` (Standard: 60 Sekunden) aktualisiert. Wenn das Element deaktiviert ist, aktualisiert der Client die Cluster-Topologie nur, falls es bei dem Versuch, Befehle für den Cluster auszuführen, zu Fehlern kommt.

Wenn diese Option aktiviert ist, können Sie die mit der Aktualisierung der Cluster-Topologie verbundene Latenz reduzieren, indem Sie diesen Auftrag zu einer Hintergrundaufgabe hinzufügen. Die Aktualisierung der Topologie erfolgt zwar im Hintergrund, kann jedoch bei Clustern mit vielen Knoten etwas langsam sein. Dies liegt daran, dass alle Knoten nach ihren Ansichten abgefragt werden, um die aktuellste Cluster-Ansicht zu erhalten. Wenn Sie einen großen Cluster betreiben, sollten Sie den Zeitraum erhöhen.

Aktivieren. [enableAllAdaptiveRefreshTriggers](#) Dadurch wird die adaptive Topologieaktualisierung aktiviert, die alle [Trigger](#) verwendet: `MOVED ASK _ REDIRECTREDIRECT`, `PERSISTENT _RECONNECTS`, `UNCOVERED _SLOT`, `UNKNOWN _NODE`. Adaptive Aktualisierungsauslöser initiieren Aktualisierungen der Topologieansicht auf der Grundlage von Ereignissen, die während Valkey- oder OSS Redis-Clustervorgängen auftreten. Das Aktivieren dieser Option führt zu einer sofortigen Aktualisierung der Topologie, wenn einer der zuvor genannten Trigger auftritt. Die Rate der adaptiv ausgelöste Aktualisierungen ist durch einen Timeout begrenzt, da Ereignisse in großem Umfang auftreten können (Standard-Timeout zwischen Updates: 30).

[closeStaleConnections](#) Aktivieren. Dadurch können alte Verbindungen beim Aktualisieren der Cluster-Topologie geschlossen werden. Es tritt nur in Kraft, wenn [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled \(\)](#) ist wahr. Wenn es aktiviert ist, kann der Client alte Verbindungen schließen und im Hintergrund neue erstellen. Dadurch wird die Notwendigkeit verringert, während der Befehlslaufzeit ausgefallene Verbindungen zu behandeln.

Aktiviert [dynamicRefreshResources](#). Wir empfehlen, sie `dynamicRefreshResources` für kleine Cluster zu aktivieren und für große Cluster zu deaktivieren. `dynamicRefreshResources` ermöglicht die Erkennung von Clusterknoten anhand des bereitgestellten Seed-Knotens (z. B. vom Cluster-Konfigurationsendpunkt). Es verwendet alle erkannten Knoten als Quellen für die Aktualisierung der Cluster-Topologie.

Bei der Verwendung der dynamischen Aktualisierung werden alle erkannten Knoten nach der Cluster-Topologie abgefragt und es wird versucht, die genaueste Cluster-Ansicht auszuwählen. Wenn der Wert auf „falsch“ festgelegt ist, werden nur die anfänglichen Seed-Knoten als Quellen für die Topologieerkennung verwendet, und die Anzahl der Clients wird nur für die ersten Seed-Knoten ermittelt. Ist er deaktiviert und der Endpunkt der Cluster-Konfiguration wird in einen ausgefallenen Knoten aufgelöst, schlägt der Versuch, die Cluster-Ansicht zu aktualisieren, fehl und führt zu Ausnahmen. Dieses Szenario kann eintreten, da es einige Zeit dauert, bis der Eintrag eines ausgefallenen Knotens vom Endpunkt der Cluster-Konfiguration entfernt wird. Daher kann der Konfigurationsendpunkt für einen kurzen Zeitraum immer noch nach dem Zufallsprinzip auf einen ausgefallenen Knoten aufgelöst werden.

Ist er aktiviert, verwenden wir jedoch alle Cluster-Knoten, die von der Cluster-Ansicht empfangen werden, um ihre aktuelle Ansicht abzufragen. Da wir ausgefallene Knoten aus dieser Ansicht herausfiltern, ist die Topologieaktualisierung erfolgreich. Wenn dies jedoch `dynamicRefreshSources` zutrifft, fragt Lettuce alle Knoten ab, um die Clusteransicht abzurufen, und vergleicht dann die Ergebnisse. Daher kann dieses Vorgehen für Cluster mit vielen Knoten teuer sein. Wir empfehlen, dass Sie diese Funktion für Cluster mit vielen Knoten deaktivieren.

```
final ClusterTopologyRefreshOptions topologyOptions =
 ClusterTopologyRefreshOptions.builder()
 .enableAllAdaptiveRefreshTriggers()
 .enablePeriodicRefresh()
 .dynamicRefreshSources(true)
 .build();
```

## ClientResources

Konfigurieren Sie [DnsResolver](#) mit [DirContextDnsResolver](#). Der DNS Resolver basiert auf `Java.com.sun.jndi.dns.DnsContextFactory`.

Konfiguration [reconnectDelay](#) mit exponentiellem Backoff und vollem Jitter. Lettuce verfügt über integrierte Wiederholungsmechanismen, die auf den exponentiellen Backoff-Strategien basieren. Einzelheiten finden Sie unter [Exponential Backoff](#) and Jitter im Architecture-Blog. AWS [Weitere](#)

[Informationen darüber, wie wichtig es ist, eine Backoff-Strategie für Wiederholungsversuche zu verwenden, finden Sie in den Abschnitten zur Backoff-Logik im Blogbeitrag Best Practices im Datenbank-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();
```

## Timeouts

Verwenden Sie einen niedrigeren Wert für das Verbindungs-Timeout als für das Befehls-Timeout. Lettuce verwendet einen verzögerten Verbindungsaufbau. Wenn also das Verbindungs-Timeout höher als das Befehls-Timeout ist, kann es nach einer Topologieaktualisierung zu einem Zeitraum mit anhaltendem Ausfall kommen, falls Lettuce versucht, eine Verbindung mit einem fehlerhaften Knoten herzustellen, und das Befehls-Timeout immer überschritten wird.

Verwenden Sie ein dynamisches Befehls-Timeout für verschiedene Befehle. Wir empfehlen, das Befehls-Timeout auf der Grundlage der erwarteten Dauer des Befehls festzulegen. Verwenden Sie beispielsweise ein längeres Timeout für Befehle, die über mehrere Schlüssel iterieren, wie `FLUSHDB`, `FLUSHALL`, `KEYS`, `SMEMBERS`. Verwenden Sie kürzere Timeouts für Befehle mit nur einer Taste, wie `SET`, und `GET`, `HSET`.

### Note

Die im folgenden Beispiel konfigurierten Timeouts gelten für Tests, bei denen `SET` `GET` /-Befehle mit Tasten und Werten von bis zu 20 Byte Länge ausgeführt wurden. Die Verarbeitungszeit kann bei komplexen Befehlen oder größeren Schlüsseln und Werten länger sein. Sie sollten die Timeouts je nach Anwendungsfall Ihrer Anwendung festlegen.

```
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();
```

## Beispiel: Lettuce-Konfiguration für den Clustermodus, aktiviert TLS

### Note

Timeouts im folgenden Beispiel beziehen sich auf Tests, bei denen SET GET /-Befehle mit Schlüsseln und Werten von bis zu 20 Byte Länge ausgeführt wurden. Die Verarbeitungszeit kann bei komplexen Befehlen oder größeren Schlüsseln und Werten länger sein. Sie sollten die Timeouts je nach Anwendungsfall Ihrer Anwendung festlegen.

```
// Set DNS cache TTL
public void setJVMProperties() {
 java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
cluster configuration endpoint
final RedisURI redisUriCluster =
 RedisURI.Builder.redis(clusterConfigurationEndpoint)
 .withPort(6379)
 .withSsl(true)
 .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
 .reconnectDelay(
 Delay.fullJitter(
 Duration.ofMillis(100), // minimum 100 millisecond delay
 Duration.ofSeconds(10), // maximum 10 second delay
 100, TimeUnit.MILLISECONDS)) // 100 millisecond base
 .dnsResolver(new DirContextDnsResolver())
 .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
 RedisClusterClient.create(clientResources, redisUriCluster);
```

```
// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
 private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
ImmutableSet.<ProtocolKeyword>builder()
 .add(CommandType.FLUSHDB)
 .add(CommandType.FLUSHALL)
 .add(CommandType.CLUSTER)
 .add(CommandType.INFO)
 .add(CommandType.KEYS)
 .build();

 private final Duration metaCommandTimeout;
 private final Duration defaultCommandTimeout;

 DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
 {
 defaultCommandTimeout = defaultTimeout;
 metaCommandTimeout = metaTimeout;
 }

 @Override
 public long getTimeout(RedisCommand<?, ?, ?> command) {
 if (META_COMMAND_TYPES.contains(command.getType())) {
 return metaCommandTimeout.toMillis();
 }
 return defaultCommandTimeout.toMillis();
 }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
 .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
META_COMMAND_TIMEOUT))
 .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
 ClusterTopologyRefreshOptions.builder()
 .enableAllAdaptiveRefreshTriggers()
 .enablePeriodicRefresh()
 .dynamicRefreshSources(true)
 .build();
```

```
// Configure the socket options
final SocketOptions socketOptions =
 SocketOptions.builder()
 .connectTimeout(CONNECT_TIMEOUT)
 .keepAlive(true)
 .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
 ClusterClientOptions.builder()
 .topologyRefreshOptions(topologyOptions)
 .socketOptions(socketOptions)
 .autoReconnect(true)
 .timeoutOptions(timeoutOptions)
 .nodeFilter(it ->
 ! (it.is(RedisClusterNode.NodeFlag.FAIL)
 || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
 || it.is(RedisClusterNode.NodeFlag.NOADDR)))
 .validateClusterNodeMembership(false)
 .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
 redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();
```

Beispiel: Die Lettuce-Konfiguration für den Cluster-Modus ist deaktiviert, aktiviert TLS

### Note

Timeouts im folgenden Beispiel beziehen sich auf Tests, bei denen SET GET /-Befehle mit Schlüsseln und Werten von bis zu 20 Byte Länge ausgeführt wurden. Die Verarbeitungszeit kann bei komplexen Befehlen oder größeren Schlüsseln und Werten länger sein. Sie sollten die Timeouts je nach Anwendungsfall Ihrer Anwendung festlegen.

```
// Set DNS cache TTL
```

```
public void setJVMProperties() {
 java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the primary/reader endpoint
clusterEndpoint = <primary/reader-endpoint> // TODO: add your node endpoint
RedisURI redisUriStandalone =

 RedisURI.Builder.redis(clusterEndpoint).withPort(6379).withSsl(true).withDatabase(0).build();

ClientResources clientResources =
 DefaultClientResources.builder()
 .dnsResolver(new DirContextDnsResolver())
 .reconnectDelay(
 Delay.fullJitter(
 Duration.ofMillis(100), // minimum 100 millisecond delay
 Duration.ofSeconds(10), // maximum 10 second delay
 100,
 TimeUnit.MILLISECONDS)) // 100 millisecond base
 .build();

// Use a dynamic timeout for commands, to avoid timeouts during
// slow operations.
class DynamicTimeout extends TimeoutSource {
 private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
 ImmutableSet.<ProtocolKeyword>builder()
 .add(CommandType.FLUSHDB)
 .add(CommandType.FLUSHALL)
 .add(CommandType.INFO)
 .add(CommandType.KEYS)
 .build();

 private final Duration metaCommandTimeout;
 private final Duration defaultCommandTimeout;

 DynamicTimeout(Duration defaultTimeout, Duration metaTimeout)
 {
 defaultCommandTimeout = defaultTimeout;
 metaCommandTimeout = metaTimeout;
 }
}
```

```

 }

 @Override
 public long getTimeout(RedisCommand<?, ?, ?> command) {
 if (META_COMMAND_TYPES.contains(command.getType())) {
 return metaCommandTimeout.toMillis();
 }
 return defaultCommandTimeout.toMillis();
 }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
 .timeoutSource(new DynamicTimeout(DEFAULT_COMMAND_TIMEOUT, META_COMMAND_TIMEOUT))
 .build();

final SocketOptions socketOptions =
 SocketOptions.builder().connectTimeout(CONNECT_TIMEOUT).keepAlive(true).build();

ClientOptions clientOptions =

 ClientOptions.builder().timeoutOptions(timeoutOptions).socketOptions(socketOptions).build();

RedisClient redisClient = RedisClient.create(clientResources, redisUriStandalone);
redisClient.setOptions(clientOptions);

```

## Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster (Valkey und Redis) OSS

Bei Valkey- oder Redis-Clustern, die im OSS Clustermodus aktiviert sind, können Sie mit dem IP Discovery-Parameter steuern, welches Protokoll Clients verwenden, um eine Verbindung zu den Knoten im Cluster herzustellen. Der IP Discovery-Parameter kann entweder auf `IPv4` oder `IPv6` festgelegt werden.

Für Valkey- oder OSS Redis-Cluster legt der IP-Discovery-Parameter das IP-Protokoll fest, das in der Ausgabe von [Cluster-Slots \(\)](#), [Cluster-Shards \(\)](#) und [Clusterknoten \(\)](#) verwendet wird. Diese Befehle werden von Clients verwendet, um die Cluster-Topologie zu ermitteln. Clients verwenden die IPs in diesen Befehlen enthaltenen Befehle, um eine Verbindung zu den anderen Knoten im Cluster herzustellen.

Eine Änderung vom IP-Discovery führt zu keinen Ausfallzeiten für verbundene Clients. Die Verarbeitung und Weiterleitung der Änderungen wird jedoch einige Zeit dauern. Um festzustellen, wann sich die Änderungen für einen Valkey- oder OSS Redis-Cluster vollständig verbreitet

haben, überwachen Sie die Ausgabe von `cluster slots`. Sobald alle vom Befehl `Cluster-Slots` zurückgegebenen Knoten das neue Protokoll gemeldet IPs haben, ist die Übertragung der Änderungen abgeschlossen.

Beispiel mit 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)
```

Beispiel mit Lettuce:

```
RedisClusterClient clusterClient = RedisClusterClient.create(RedisURI.create("xxxx",
6379));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

do {
 // Check for any changes in the cluster topology.
 // Under the hood this calls cluster slots
 clusterClient.refreshPartitions();
 Set<String> nodes = new HashSet<>();

 for (RedisClusterNode node : clusterClient.getPartitions().getPartitions()) {
 nodes.add(node.getUri().getHost());
 }
}
```

```
Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
 try {
 return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
 } catch (UnknownHostException ignored) {}
 return false;
}));
```

## Bewährte Methoden für Kunden (Memcached)

### Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich (Memcached)

#### Note

Dieser Abschnitt trifft auf selbst entworfene Memcached-Cluster mit mehreren Knoten zu.

Um mehrere ElastiCache Memcached-Knoten effektiv nutzen zu können, müssen Sie in der Lage sein, Ihre Cache-Schlüssel auf die Knoten zu verteilen. Eine einfache Methode für den Lastenausgleich eines Clusters mit  $n$  Knoten besteht in der Berechnung des Hash-Wertes des Objektschlüssels und Verschiebung des Ergebnisses um  $n - \text{hash}(\text{key}) \bmod n$ . Der resultierende Wert (0 bis  $n-1$ ) ist die Nummer des Knotens, auf dem Sie das Objekt platzieren.

Diese Methode ist einfach und funktioniert gut, solange die Anzahl der Knoten ( $n$ ) konstant bleibt. Wenn Sie einen Knoten zum Cluster hinzufügen oder daraus entfernen, beträgt die Anzahl der zu verschiebenden Schlüssel jedoch jedes Mal  $(n - 1) / n$  (dabei entspricht  $n$  der neuen Anzahl von Knoten). Diese Methode führt daher dazu, dass eine große Anzahl von Schlüsseln verschoben werden. Dies ist mit einer großen Anzahl anfänglicher Cache-Fehlelemente verbunden, insbesondere bei einer umfangreichen Anzahl von Knoten. Die Skalierung von 1 auf 2 Knoten führt bestenfalls dazu, dass  $(2-1) / 2$  (50 %) der Schlüssel verschoben werden. Eine Skalierung von 9 auf 10 Knoten hat bestenfalls zur Folge, dass  $(10-1) / 10$  (90 %) der Schlüssel verschoben werden. Ist die Aufwärtsskalierung durch einen sprunghaften Anstieg im Datenverkehr bedingt, ist eine große Anzahl von Cache-Fehlelementen nicht wünschenswert. Eine große Anzahl von Cache-Fehlelementen führt zu Anfragen bei der Datenbank, die aufgrund des sprunghaften Anstiegs im Datenverkehr bereits überlastet ist.

Die Lösung zu diesem Dilemma ist konsistentes Hashing. Beim konsistenten Hashing wird ein Algorithmus verwendet, bei dem jedes Mal, wenn ein Knoten zu einem Cluster hinzugefügt oder

daraus entfernt wird, die Anzahl der zu verschiebenden Schlüssel ungefähr  $1/n$  beträgt (wobei  $n$  die neue Anzahl von Knoten ist). Eine Skalierung von 1 auf 2 Knoten hat schlimmstenfalls zur Folge, dass  $1/2$  (50 Prozent) der Schlüssel verschoben werden. Eine Skalierung von 9 auf 10 Knoten hat zur Folge, dass  $1/10$  (10 Prozent) der Schlüssel verschoben werden.

Sie als Benutzer steuern, welcher Hashalgorithmus für Mehrknoten-Cluster verwendet wird. Wir empfehlen, dass Sie Ihre Clients zur Verwendung von konsistentem Hashing konfigurieren. Glücklicherweise sind viele Memcached-Clientbibliotheken in den meisten gängigen Sprachen verfügbar, die konsistentes Hashing implementieren. Überprüfen Sie die Dokumentation der von Ihnen verwendeten Bibliothek darauf, ob sie konsistentes Hashing unterstützt und wie es implementiert wird.

Wenn Sie in Java arbeiten PHP, oder .NET, wir empfehlen Ihnen, eine der ElastiCache Amazon-Clientbibliotheken zu verwenden.

### Konsistentes Hashing mithilfe von Java

Der ElastiCache Memcached-Java-Client basiert auf dem Open-Source-Spymemcached-Java-Client, der über integrierte konsistente Hashing-Funktionen verfügt. Die Bibliothek umfasst eine Klasse, die konsistentes Hashing implementiert. `KetamaConnectionFactory` Konsistentes Hashing ist in `spymemcached` standardmäßig deaktiviert.

Weitere Informationen finden Sie in der `KetamaConnectionFactory` Dokumentation unter [KetamaConnectionFactory](#).

### Konsistentes Hashing unter Verwendung von PHP Memcached

Der ElastiCache PHP Memcached-Client ist ein Wrapper rund um die integrierte Memcached-Bibliothek. PHP Standardmäßig ist konsistentes Hashing von der Memcached-Bibliothek deaktiviert.

PHP

Verwenden Sie den folgenden Code, um konsistentes Hashing zu aktivieren.

```
$m = new Memcached();
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

Zusätzlich zum obigen Code wird empfohlen, ebenfalls `memcached.sess_consistent_hash` in der Datei `php.ini` einzuschalten.

Weitere Informationen finden Sie in der Dokumentation zur Laufzeitkonfiguration für Memcached unter <http://php.net/manual/en/memcached.configuration.php>. PHP Beachten Sie insbesondere den Parameter `memcached.sess_consistent_hash`.

Konsistentes Hashing unter Verwendung von NET mit Memcached

Der ElastiCache Memcached.NET Client ist ein Wrapper rund um Enyim Memcached. Konsistentes Hashing wird vom Enyim-Memcached-Client standardmäßig aktiviert.

Weitere Informationen finden Sie in der Dokumentation unter [/wiki/-Configuration#memcached/locator](#). <https://github.com/enyim/EnyimMemcached> MemcachedClient user-content-memcachedlocator

## Validierte Kunden mit Memcached

Die folgenden Clients wurden speziell daraufhin geprüft, ob sie mit allen unterstützten Netzwerktopologien für Memcached funktionieren.

Validierte Clients:

- [AWS ElastiCache Cluster-Client Memcached für Php — Version \\*3.6.2](#)
- [AWS ElastiCache Cluster Client Memcached für Java](#) — Neuester Master auf Github

## Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster (Memcached)

Bei Memcached-Clustern können Sie mit dem IP-Discovery-Parameter das Protokoll steuern, das Clients für die Verbindung mit den Knoten im Cluster verwenden. Der IP Discovery-Parameter kann entweder auf `IPv4` oder `IPv6` gesetzt werden.

Der IP-Erkennungsparameter steuert das IP-Protokoll, das in der Cluster-Ausgabe der Konfiguration verwendet wird. Dies wiederum bestimmt das IP-Protokoll, das von Clients verwendet wird, die Autodiscovery für ElastiCache (Memcached-) Cluster unterstützen.

Eine Änderung vom IP-Discovery führt zu keinen Ausfallzeiten für verbundene Clients. Die Weiterleitung der Änderungen wird jedoch einige Zeit dauern.

Überwachen Sie die Ausgabe von `getAvailableNodeEndpoints` für Java und für Php überwachen Sie die Ausgabe von `getServerList`. Sobald die Ausgabe dieser Funktionen Berichte IPs für alle Knoten im Cluster enthält, die das aktualisierte Protokoll verwenden, ist die Übertragung der Änderungen abgeschlossen.

## Java-Beispiel:

```

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;
}));

```

## Php-Beispiel:

```

$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);
```

Alle vorhandenen Client-Verbindungen, die vor der Aktualisierung von IP Discovery erstellt wurden, werden weiterhin mit dem alten Protokoll verbunden. Alle validierten Clients stellen mithilfe des neuen IP-Protokolls automatisch wieder eine Verbindung mit dem Cluster her, sobald die Änderungen in der Ausgabe der Cluster-Erkennungsbefehle erkannt wurden. Dies hängt jedoch von der Implementierung des Clients ab.

## TLSaktivierte ElastiCache Dual-Stack-Cluster

Wenn für ElastiCache Cluster aktiviert TLS ist, geben die Cluster-Erkennungsfunktionen (`cluster slotscluster shards`, und `cluster nodes` für Redis) oder `config get cluster` für Memcached statt Hostnamen zurück. IPs Die Hostnamen werden dann verwendet, anstatt eine Verbindung IPs zum ElastiCache Cluster herzustellen und einen Handshake durchzuführen. TLS Das bedeutet, dass Clients nicht vom IP-Discovery-Parameter betroffen sind. Bei TLS aktivierten Clustern hat der IP Discovery-Parameter keine Auswirkung auf das bevorzugte IP-Protokoll. Stattdessen wird das verwendete IP-Protokoll davon bestimmt, welches IP-Protokoll der Client bei der Auflösung von DNS Hostnamen bevorzugt.

### Java-Clients

Wenn Sie eine Verbindung von einer Java-Umgebung aus herstellen, die IPv4 sowohl als auch unterstützt IPv6, wird Java aus IPv6 Gründen der Abwärtskompatibilität standardmäßig IPv4 den Vorzug geben. Die IP-Protokollpräferenz kann jedoch über die JVM Argumente konfiguriert werden. Zu bevorzugen IPv4, zu JVM akzeptieren `-Djava.net.preferIPv4Stack=true` und zu IPv6 bevorzugen `-Djava.net.preferIPv6Stack=true`. Die Einstellung `-Djava.net.preferIPv4Stack=true` bedeutet, JVM dass keine IPv6 Verbindungen mehr hergestellt werden. Bei Valkey oder Redis umfasst dies auch die OSS Verbindungen zu anderen Nicht-Valkey- und Nicht-Redis-Anwendungen. OSS

### Einstellungen auf Host-Ebene

Wenn der Client oder die Client-Laufzeit keine Konfigurationsoptionen für die Einstellung einer IP-Protokollpräferenz bereitstellt, hängt das IP-Protokoll bei der DNS Auflösung generell von der Konfiguration des Hosts ab. Standardmäßig bevorzugen die meisten Hosts IPv6 gegenüber, IPv4 aber diese Präferenz kann auf Host-Ebene konfiguriert werden. Dies wirkt sich auf alle DNS Anfragen von diesem Host aus, nicht nur auf Anfragen an ElastiCache Cluster.

## Linux-Hosts

Für Linux kann eine IP-Protokollpräferenz konfiguriert werden, indem die `gai.conf`-Datei geändert wird. Sie finden die `gai.conf`-Datei unter `/etc/gai.conf`. Wenn keine `gai.conf` angegeben ist, sollte ein Beispiel davon `/usr/share/doc/glibc-common-x.xx/gai.conf` verfügbar sein, das nach `/etc/gai.conf` kopiert werden kann. Die Standardkonfiguration sollte dann unkommentiert sein. Um die Konfiguration zu aktualisieren, die IPv4 bei der Verbindung zu einem ElastiCache Cluster bevorzugt wird, aktualisieren Sie die Priorität für den CIDR Bereich, der den Cluster umfasst, soIPs, dass sie über der Priorität für Standardverbindungen liegt. IPv6 Standardmäßig haben IPv6 Verbindungen eine Priorität von 40. Angenommen, der Cluster befindet sich in einem Subnetz mit CIDR `172.31.0.0/16`, würde die folgende Konfiguration dazu führen, dass Clients Verbindungen zu diesem Cluster bevorzugen. 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
```

Weitere Informationen zu `gai.conf` finden Sie auf der [Linux-Hauptseite](#).

## Windows-Hosts

Der Prozess für Windows-Hosts ist ähnlich. Für Windows-Hosts können Sie `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL` ausführen. Dies hat den gleichen Effekt wie das Ändern der `gai.conf`-Datei für Linux-Hosts.

Dadurch werden die Präferenzrichtlinien aktualisiert, sodass IPv4 Verbindungen für den angegebenen Bereich vorgezogen werden. Wenn beispielsweise angenommen wird, dass sich der Cluster in einem Subnetz befindet, in dem `172.31.0.0:0/16` CIDR ausgeführt wird, `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` würde dies zu der folgenden Rangfolgetabelle führen, was dazu führen würde, dass Clients eine Verbindung zum Cluster bevorzugen. 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
```

## Verwaltung des reservierten Speichers für Valkey und Redis OSS

Reservierter Speicher ist Speicher, der nicht für Daten verwendet werden soll. Bei der Durchführung eines Backups oder Failovers OSS verwenden Valkey und Redis den verfügbaren Speicher, um Schreibvorgänge in Ihrem Cluster aufzuzeichnen, während die Daten des Clusters in die RDB-Datei geschrieben werden. Wenn nicht genügend Speicher für alle Schreibvorgängen zur Verfügung steht, tritt ein Fehler auf. Im Folgenden finden Sie Informationen zu Optionen für die Verwaltung von reserviertem Speicher für ElastiCache (RedisOSS) und zur Anwendung dieser Optionen.

### Themen

- [Wie viel reservierten Speicher benötigen Sie?](#)
- [Parameter zum Veralten von reserviertem Speicher](#)
- [Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers](#)

### Wie viel reservierten Speicher benötigen Sie?

Wenn Sie eine Version von Redis OSS vor 2.8.22 ausführen, reservieren Sie mehr Speicher für Backups und Failover, als wenn Sie Redis 2.8.22 oder höher ausführen. OSS Diese Anforderung ist auf die unterschiedlichen Methoden zurückzuführen, mit denen ElastiCache (Redis) den Backup-Prozess implementiert. OSS Als Faustregel gilt, die Hälfte des `maxmemory` Werts eines Knotentyps für OSS Redis-Overhead für Versionen vor 2.8.22 und ein Viertel für OSS Redis-Versionen 2.8.22 und höher zu reservieren.

Aufgrund der unterschiedlichen ElastiCache Implementierungsmethoden des Sicherungs- und Replikationsprozesses lautet die Faustregel, 25% des Werts eines Knotentyps mithilfe des Parameters zu reservieren. `maxmemory reserved-memory-percent` Dies ist der Standardwert und wird in den meisten Fällen empfohlen.

Wenn die Typen Burstable Micro und Small Instances in der Nähe der `maxmemory` Grenzwerte arbeiten, kann es zu Swap-Nutzung kommen. Um die Betriebssicherheit dieser Instance-Typen bei Backups, Replikationen und hohem Datenaufkommen zu verbessern, empfehlen wir, den Wert des `reserved-memory-percent` Parameters bei kleinen Instance-Typen um bis zu 30% und bei Micro-Instance-Typen um bis zu 50% zu erhöhen.

Für schreibintensive Workloads auf ElastiCache Clustern mit Daten-Tiering empfehlen wir, den verfügbaren Speicher des Knotens auf `reserved-memory-percent` bis zu 50% zu erhöhen.

Weitere Informationen finden Sie hier:

- [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Daten-Tiering ElastiCache](#)

## Parameter zum Veralten von reserviertem Speicher

Seit dem 16. März 2017 ElastiCache bietet Amazon zwei sich gegenseitig ausschließende Parameter für die Verwaltung Ihres Valkey- oder OSS Redis-Speichers, `undreserved-memory`. `reserved-memory-percent` Keiner dieser Parameter ist Teil der Valkey- oder Redis-Distribution. OSS

Je nachdem, wann Sie ElastiCache Kunde wurden, ist der eine oder andere dieser Parameter der Standardparameter für die Speicherverwaltung. Dieser Parameter gilt, wenn Sie einen neuen Valkey- oder OSS Redis-Cluster oder eine neue Replikationsgruppe erstellen und eine Standardparametergruppe verwenden.

- Für Kunden, die vor dem 16. März 2017 angefangen haben — Wenn Sie einen OSS Redis-Cluster oder eine Redis-Replikationsgruppe mithilfe der Standardparametergruppe erstellen, lautet Ihr Speicherverwaltungsparameter `reserved-memory` In diesem Fall werden null (0) Byte Speicher reserviert.
- Für Kunden, die am oder nach dem 16. März 2017 angefangen haben: Wenn Sie einen Valkey- oder OSS Redis-Cluster oder eine Replikationsgruppe mithilfe der Standardparametergruppe erstellen, lautet Ihr Speicherverwaltungsparameter `reserved-memory-percent` In diesem Fall sind 25 Prozent des `maxmemory`-Wertes Ihres Knotens für Nichtdatenzwecke reserviert.

Nachdem Sie sich mit den beiden OSS Speicherverwaltungsparametern Valkey oder Redis vertraut gemacht haben, ziehen Sie es vielleicht vor, den zu verwenden, der nicht Ihre Standardwerte ist oder nicht die Standardwerte enthält. Wenn dies der Fall ist, können Sie zu dem anderen Verwaltungsparameter für reservierten Speicher wechseln.

Um den Wert dieses Parameters zu ändern, können Sie eine benutzerdefinierte Parametergruppe erstellen und sie ändern, um Ihren bevorzugten Speicherverwaltungsparameter und -wert zu verwenden. Sie können die benutzerdefinierte Parametergruppe dann immer dann verwenden, wenn Sie einen neuen Valkey- oder OSS Redis-Cluster oder eine neue Replikationsgruppe erstellen. Existierende Cluster und Replikationsgruppen können Sie so ändern, dass sie die benutzerdefinierte Parametergruppe verwenden.

Weitere Informationen finden Sie hier:

- [Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers](#)
- [Eine ElastiCache Parametergruppe erstellen](#)
- [Änderung einer ElastiCache Parametergruppe](#)
- [Einen ElastiCache Cluster ändern](#)
- [Ändern einer Replikationsgruppe](#)

Der Parameter "reserved-memory"

Vor dem 16. März 2017 wurde die gesamte Verwaltung des reservierten Speichers ElastiCache (RedisOSS) mithilfe des Parameters durchgeführt. `reserved-memory` Der Standardwert von `reserved-memory` ist 0. Diese Standardeinstellung reserviert keinen Speicher für Valkey- oder OSS Redis-Overhead und ermöglicht es Valkey oder Redis, den gesamten Speicher eines Knotens mit Daten OSS zu verbrauchen.

Wenn Sie `reserved-memory` ändern, damit ausreichend Speicher für Sicherungen und Failovers zur Verfügung steht, müssen Sie eine benutzerdefinierte Parametergruppe erstellen. In dieser benutzerdefinierten Parametergruppe legen Sie einen Wert fest, der für `reserved-memory` die Valkey- oder OSS Redis-Version, die auf Ihrem Cluster ausgeführt wird, und für den Knotentyp des Clusters geeignet ist. Weitere Informationen finden Sie unter [Wie viel reservierten Speicher benötigen Sie?](#)

Der Parameter `reserved-memory` ist spezifisch für die allgemeine OSS Redis-Distribution ElastiCache und nicht Teil dieser.

Das folgende Verfahren zeigt, wie Sie `reserved-memory` den Speicher auf Ihrem Valkey- oder OSS Redis-Cluster verwalten können.

So reservieren Sie Speicher mithilfe von reserviertem Speicher

1. Erstellen Sie eine benutzerdefinierte Parametergruppe, indem Sie die Parametergruppenfamilie angeben, die der von Ihnen ausgeführten Engine-Version entspricht, z. B. indem Sie die `redis2.8`-Parametergruppenfamilie angeben. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name redis6x-m3x1 \
 --parameter-name reserved-memory 100
```

```
--description "Redis OSS 2.8.x for m3.xlarge node type" \
--cache-parameter-group-family redis6.x
```

2. Berechnen Sie, wie viele Byte Speicher für den Valkey- oder Redis-Overhead reserviert werden müssen. OSS Sie finden den Wert von `maxmemory` für Ihren Knotentyp unter [OSSRedis-Knotentyp-spezifische Parameter](#).
3. Ändern Sie die benutzerdefinierte Parametergruppe, sodass der Parameter `reserved-memory` der Anzahl von Bytes entspricht, die Sie im vorherigen Schritt berechnet haben. Im folgenden AWS CLI Beispiel wird davon ausgegangen, dass Sie eine Version von Redis OSS vor 2.8.22 ausführen und die Hälfte der Knoten reservieren müssen. `maxmemory` Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis28-m3x1 \
 --parameter-name-values "ParameterName=reserved-memory,
 ParameterValue=7130316800"
```

Sie benötigen für jeden verwendeten Knotentyp eine eigene benutzerdefinierte Parametergruppe, da jeder Knotentyp einen anderen `maxmemory`-Wert hat. Daher benötigt jeder Knotentyp einen anderen Wert für `reserved-memory`.

4. Ändern Sie Ihren OSS Redis-Cluster oder Ihre Redis-Replikationsgruppe so, dass sie Ihre benutzerdefinierte Parametergruppe verwendet.

Im folgenden CLI Beispiel wird der Cluster so geändert `my-redis-cluster`, dass er `redis28-m3x1` ab sofort die benutzerdefinierte Parametergruppe verwendet. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-redis-cluster \
 --cache-parameter-group-name redis28-m3x1 \
 --apply-immediately
```

Im folgenden CLI Beispiel wird die Replikationsgruppe so geändert `my-redis-repl-grp`, dass sie `redis28-m3x1` ab sofort die benutzerdefinierte Parametergruppe verwendet. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

```
aws elasticache modify-replication-group \
 --replication-group-id my-redis-repl-grp \
 --cache-parameter-group-name redis28-m3x1 \
 --apply-immediately
```

```
--apply-immediately
```

## Der Parameter `reserved-memory-percent`

Am 16. März 2017 ElastiCache führte Amazon den Parameter ein `reserved-memory-percent` und stellte ihn in allen Versionen von ElastiCache (RedisOSS) zur Verfügung. Der Zweck von `reserved-memory-percent` besteht darin, die Verwaltung des reservierten Speichers für alle Ihre Cluster zu vereinfachen. Dies wird dadurch erreicht, dass Sie eine einzige Parametergruppe für jede Parametergruppenfamilie (wie `redis2.8`) zur Verfügung haben, um den reservierten Speicher Ihrer Cluster unabhängig vom Knotentyp zu verwalten. Der Standardwert für `reserved-memory-percent` ist 25 (25 Prozent).

Der Parameter `reserved-memory-percent` ist spezifisch für die allgemeine OSS Redis-Distribution ElastiCache und nicht Teil dieser.

Wenn Ihr Cluster einen Knotentyp aus der R6gd-Familie verwendet und Ihre Speicherauslastung 75 Prozent erreicht, wird Daten-Tiering automatisch ausgelöst. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

## Um Speicher zu reservieren mit `reserved-memory-percent`

Gehen Sie wie folgt vor, `reserved-memory-percent` um den Speicher auf Ihrem ElastiCache (Redis-OSS) Cluster zu verwalten:

- Wenn Sie Redis OSS 2.8.22 oder höher ausführen, weisen Sie Ihrem Cluster die Standardparametergruppe zu. Der Standardwert 25 Prozent sollte ausreichen. Falls nicht, führen Sie die folgenden Schritte aus, um den Wert zu ändern.
- Wenn Sie eine Version von Redis OSS vor 2.8.22 ausführen, müssen Sie wahrscheinlich mehr Speicher als die standardmäßigen 25 Prozent reservieren. `reserved-memory-percent` Führen Sie dazu die folgenden Schritte aus.

## Um den Prozentwert von zu ändern `reserved-memory-percent`

1. Erstellen Sie eine benutzerdefinierte Parametergruppe, indem Sie die Parametergruppenfamilie angeben, die der von Ihnen ausgeführten Engine-Version entspricht, z. B. indem Sie die `redis2.8`-Parametergruppenfamilie angeben. Eine benutzerdefinierte Parametergruppe ist erforderlich, da Sie eine Standardparametergruppe nicht ändern können. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

```
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
```

Da `reserved-memory-percent` Speicher als Prozentanteil von `maxmemory` eines Knotens reserviert, benötigen Sie keine benutzerdefinierte Parametergruppe für jeden Knotentyp.

2. Ändern Sie die benutzerdefinierte Parametergruppe, sodass `reserved-memory-percent` 50 (50 Prozent) beträgt. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis28-50 \
 --parameter-name-values "ParameterName=reserved-memory-percent,
 ParameterValue=50"
```

3. Verwenden Sie diese benutzerdefinierte Parametergruppe für alle OSS Redis-Cluster oder Replikationsgruppen, auf denen eine Version von Redis ausgeführt wird, die OSS älter als 2.8.22 ist.

Im folgenden CLI Beispiel wird der OSS Redis-Cluster so geändert, dass er ab sofort die `my-redis-cluster` benutzerdefinierte Parametergruppe verwendet. `redis28-50` Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-redis-cluster \
 --cache-parameter-group-name redis28-50 \
 --apply-immediately
```

Im folgenden CLI Beispiel wird die OSS Redis-Replikationsgruppe so geändert `my-redis-repl-grp`, dass sie ab sofort die benutzerdefinierte Parametergruppe verwendet. `redis28-50` Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

```
aws elasticache modify-replication-group \
 --replication-group-id my-redis-repl-grp \
 --cache-parameter-group-name redis28-50 \
 --apply-immediately
```

## Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers

Wenn Sie am 16. März 2017 ein aktueller ElastiCache Kunde waren, lautet Ihr Standardparameter für die Verwaltung von reserviertem Speicher `reserved-memory` mit null (0) Byte reserviertem Speicher. Wenn Sie nach dem 16. März 2017 ElastiCache Kunde wurden, ist `reserved-memory-percent` Ihr Standardparameter für die Verwaltung von reserviertem Speicher so, dass 25 Prozent des Speichers des Knotens reserviert sind. Dies gilt unabhängig davon, wann Sie Ihren ElastiCache (Redis-OSS) Cluster oder Ihre Replikationsgruppe erstellt haben. Sie können Ihren Parameter für die Verwaltung des reservierten Speichers jedoch entweder mit AWS CLI oder ElastiCache API ändern.

Die Parameter `reserved-memory` und `reserved-memory-percent` schließen sich gegenseitig aus. Eine Parametergruppe verfügt immer über einen dieser Parameter, niemals jedoch über beide. Sie können den Parameter ändern, den eine Parametergruppe für die Verwaltung von reserviertem Speicher verwendet, indem Sie die Parametergruppe ändern. Die Parametergruppe muss eine benutzerdefinierte Parametergruppe sein, da Sie die Standardparametergruppen nicht ändern können. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

Um zu spezifizieren `reserved-memory-percent`

Um `reserved-memory-percent` als Verwaltungsparameter für reservierten Speicher zu verwenden, ändern Sie eine benutzerdefinierte Parametergruppe mit dem `modify-cache-parameter-group`-Befehl. Verwenden Sie den `parameter-name-values`-Parameter zum Angeben von `reserved-memory-percent` sowie einen Wert dafür.

Im folgenden CLI Beispiel wird die benutzerdefinierte Parametergruppe `redis32-cluster-on` so geändert, dass sie `reserved-memory-percent` zur Verwaltung des reservierten Speichers verwendet wird. Für die Parametergruppe muss `ParameterValue` ein Wert zugewiesen werden, um den `ParameterName`-Parameter für die Verwaltung des reservierten Speichers zu verwenden. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis32-cluster-on \
 --parameter-name-values "ParameterName=reserved-memory-percent, ParameterValue=25"
```

So geben Sie `reserved-memory` an

Um `reserved-memory` als Verwaltungsparameter für reservierten Speicher zu verwenden, ändern Sie eine benutzerdefinierte Parametergruppe mit dem `modify-cache-parameter-group`-Befehl. Verwenden Sie den `parameter-name-values`-Parameter zum Angeben von `reserved-memory` sowie einen Wert dafür.

Im folgenden CLI Beispiel wird die benutzerdefinierte Parametergruppe `redis32-m3x1` so geändert, dass sie `reserved-memory` zur Verwaltung des reservierten Speichers verwendet wird. Für die Parametergruppe muss `ParameterValue` ein Wert zugewiesen werden, um den `ParameterName`-Parameter für die Verwaltung des reservierten Speichers zu verwenden. Da die Engine-Version höher als 2.8.22 ist, wird der Wert auf `3565158400` gesetzt. Dieser Wert entspricht 25 % des Werts von `maxmemory` für `cache.m3.xlarge`. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

```
aws elasticache modify-cache-parameter-group \
 --cache-parameter-group-name redis32-m3x1 \
 --parameter-name-values "ParameterName=reserved-memory, ParameterValue=3565158400"
```

## Bewährte Methoden bei der Arbeit mit selbst entworfenen Clustern von Valkey und Redis OSS

Multi-AZ-Nutzung, ausreichender Arbeitsspeicher, Clustergrößenänderung und Minimierung von Ausfallzeiten sind alles nützliche Konzepte, die Sie bei der Arbeit mit selbst entworfenen Clustern in Valkey oder Redis berücksichtigen sollten. OSS Wir empfehlen Ihnen, sich mit diesen bewährten Methode vertraut zu machen und sie zu befolgen.

### Themen

- [Minimierung von Ausfallzeiten mit Multi-AZ](#)
- [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#)
- [Online-Größenanpassung von Clustern](#)
- [Minimieren der Ausfallzeit während der Wartung](#)

### Minimierung von Ausfallzeiten mit Multi-AZ

Es gibt eine Reihe von Fällen, in denen ElastiCache Valkey oder Redis OSS möglicherweise einen Primärknoten austauschen müssen. Dazu gehören bestimmte Arten von geplanten Wartungsarbeiten und der unwahrscheinliche Fall eines Ausfalls eines Primärknotens oder einer Availability Zone.

Dieser Austausch führt zu einer gewissen Ausfallzeit für den Cluster, aber wenn Multi-AZ aktiviert ist, wird die Ausfallzeit minimiert. Die Rolle des primären Knotens wird automatisch auf eines der Read Replicas übertragen. Es ist nicht erforderlich, einen neuen Primärknoten zu erstellen und bereitzustellen, da dies transparent ElastiCache gehandhabt wird. Dieser Failover und die

Replikatheraufstufung stellen sicher, dass Sie weiter in den neuen primären Knoten schreiben können, sobald die Heraufstufung abgeschlossen wurde.

Weitere Informationen [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#) über Multi-AZ und die Minimierung von Ausfallzeiten finden Sie unter.

Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen

Schnappschüsse und Synchronisationen in Valkey 7.2 und höher sowie in Redis Version 2.8.22 und höher OSS

Valkey unterstützt standardmäßig Snapshots und Synchronisationen. Redis OSS 2.8.22 führt einen forkless-Speicherprozess ein, mit dem Sie einen größeren Teil Ihres Speichers für die Nutzung durch Ihre Anwendung zuweisen können, ohne dass bei Synchronisationen und Speichervorgängen eine erhöhte Swap-Auslastung entsteht. Weitere Informationen finden Sie unter [So werden Synchronisation und Backup implementiert](#).

Redis-Snapshots und Synchronisationen vor Version OSS 2.8.22

Wenn Sie mit ElastiCache (RedisOSS) arbeiten, OSS ruft Redis in einer Reihe von Fällen einen Schreibbefehl im Hintergrund auf:

- Beim Erstellen eines Snapshots für eine Sicherung
- Beim Synchronisieren von Replikaten mit dem primären Cluster in einer Replikationsgruppe
- Wenn Sie die Funktion „Nur Dateien anhängen“ () AOF für Redis aktivieren. OSS
- Beim Hochstufen eines Replikats zu einem primären (was eine primäre/Replikat-Synchronisierung verursacht).

Immer wenn Redis einen Schreibvorgang im Hintergrund OSS ausführt, müssen Sie über ausreichend verfügbaren Speicher verfügen, um den Prozessaufwand zu bewältigen. Wenn nicht genügend Arbeitsspeicher verfügbar ist, schlägt der Vorgang fehl. Aus diesem Grund ist es wichtig, bei der Erstellung Ihres OSS Redis-Clusters einen Knoteninstanztyp auszuwählen, der über ausreichend Speicher verfügt.

Schreibvorgang im Hintergrund und Speichernutzung mit Valkey und Redis OSS

Immer wenn ein Schreibprozess im Hintergrund aufgerufen wird, OSS forken Valkey und Redis seinen Prozess ab (denken Sie daran, dass diese Engines Single-Threading verwenden). Ein

Fork speichert Ihre Daten auf der Festplatte in einer OSS Redis-.rdb-Snapshot-Datei. Die andere Vergabelung führt alle Lese- und Schreibvorgänge durch. Um sicherzustellen, dass es sich bei Ihrem Snapshot um einen point-in-time Snapshot handelt, werden alle Datenaktualisierungen und -ergänzungen in einen vom Datenbereich getrennten Bereich des verfügbaren Speichers geschrieben.

Solange genügend Arbeitsspeicher zum Aufzeichnen aller Schreibvorgänge verfügbar ist, während die Daten dauerhaft auf dem Datenträger erhalten bleiben, treten keine Probleme aufgrund von Speichermangel auf. Wenn vermehrt Probleme aufgrund von Speichermangel auftreten, treffen beliebige der folgenden Situationen ein:

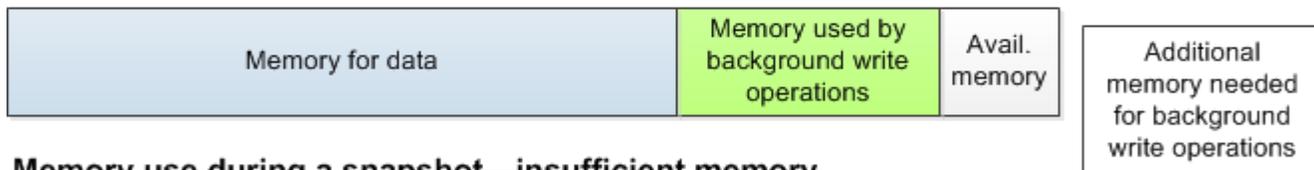
- Ihre Anwendung führt viele Schreibvorgänge aus und benötigt daher eine große Menge an verfügbarem Arbeitsspeicher zum Akzeptieren neuer oder aktualisierter Daten.
- Es ist sehr wenig Arbeitsspeicher zum Schreiben neuer oder aktualisierter Daten verfügbar.
- Die dauerhafte Erhaltung eines großen Datensatzes auf der Festplatte dauert eine lange Zeit und erfordert eine große Anzahl von Schreibvorgängen.

Das folgende Diagramm veranschaulicht die Speichernutzung beim Ausführen eines Hintergrundschreibvorgangs.

#### Memory use prior to a snapshot



#### Memory use during a snapshot—sufficient memory



#### Memory use during a snapshot—insufficient memory



Informationen zu den Auswirkungen einer Sicherung auf die Leistung finden Sie unter [Auswirkungen von Backups selbst entworfener Cluster auf die Leistung](#).

Weitere Informationen darüber, wie Valkey und Redis Snapshots OSS durchführen, finden Sie unter <http://valkey.io>.

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

## Vermeidung von Speichermangel beim Ausführen eines Hintergrundschreibvorgangs

Immer wenn ein Schreibvorgang im Hintergrund wie BGSAVE oder aufgerufen BGREWRITEAOF wird, muss mehr Speicher verfügbar sein, als durch Schreibvorgänge während des Vorgangs verbraucht wird, um zu verhindern, dass der Prozess fehlschlägt. Im schlimmsten Fall wird während des Schreibvorgangs im Hintergrund jeder Datensatz aktualisiert und einige neue Datensätze werden dem Cache hinzugefügt. Aus diesem Grund empfehlen wir, den Wert `reserved-memory-percent` auf 50 (50 Prozent) für OSS Redis-Versionen vor 2.8.22 oder 25 (25 Prozent) für Valkey und alle Redis-Versionen 2.8.22 und höher festzulegen. OSS

Der Wert `maxmemory` gibt den für die Daten und den Betriebsaufwand verfügbaren Arbeitsspeicher an. Da der Parameter `reserved-memory` in der Standardparametergruppe nicht geändert werden kann, müssen Sie eine benutzerdefinierte Parametergruppe für den Cluster erstellen. Der Standardwert für `reserved-memory` ist 0, was es Redis ermöglicht, den gesamten Maximalspeicher mit Daten OSS zu verbrauchen, sodass möglicherweise zu wenig Speicher für andere Zwecke übrig bleibt, z. B. für einen Schreibvorgang im Hintergrund. Informationen zu `maxmemory`-Werten nach Knoten-Instance-Typ finden Sie unter [OSSRedis-Knotentyp-spezifische Parameter](#).

Sie können `reserved-memory` Parameter auch verwenden, um den Speicherverbrauch auf der Box zu reduzieren.

Weitere Informationen zu Valkey- und Redis-spezifischen Parametern finden Sie unter [ElastiCache Valkey- und Redis-Parameter OSS](#)

Weitere Informationen zum Erstellen und Ändern von Parametergruppen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#) und [Änderung einer ElastiCache Parametergruppe](#).

## Online-Größenanpassung von Clustern

Resharding umfasst das Hinzufügen und Entfernen von Shards oder Knoten für den Cluster sowie die Neuverteilung von Schlüsselräumen. Daher haben viele Aspekte Einfluss auf die Resharding-Operation, z. B. Workload des Clusters, Speichernutzung und allgemeine Datengröße. Für optimale Ergebnisse empfehlen wir, dass Sie die allgemeinen bewährten Methoden zu Clustern für eine gleichmäßige Verteilung von Workload-Verteilung befolgen. Außerdem empfehlen wir, die folgenden Schritte durchzuführen.

Vor dem Beginn des Resharding sollten Sie Folgendes durchführen:

- Testen Sie Ihre Anwendung – Testen Sie das Verhalten Ihrer Anwendung während des Reshardings nach Möglichkeit in einer Staging-Umgebung.
- Erhalten Sie frühzeitige Benachrichtigungen bei Skalierungsproblemen – Resharding ist ein rechenintensiver Vorgang. Aus diesem Grund empfehlen wir, beim Resharding die CPU Auslastung bei Multicore-Instances unter 80 Prozent und bei Single-Core-Instances unter 50 Prozent zu halten. Überwachen Sie ElastiCache (Redis-OSS) Metriken und initiieren Sie das Resharding, bevor Ihre Anwendung Skalierungsprobleme beobachtet. Die Überwachung folgender Metriken ist nützlich: `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, `NewConnections`, `FreeableMemory`, `SwapUsage` und `BytesUsedForCacheItems`.
- Stellen Sie vor dem Hochskalieren sicher, dass ausreichend freier Speicher verfügbar ist – Stellen Sie beim Hochskalieren sicher, dass der freie Speicher auf den beizubehaltenden Shards mindestens das 1,5-fache des Arbeitsspeichers beträgt, der auf den Shards verwendet wird, die Sie entfernen möchten.
- Initiieren Sie Resharding außerhalb der Spitzenzeiten – Diese Vorgehensweise hilft, die Auswirkungen auf die Latenz und den Durchsatz auf den Client während des Resharding-Vorgangs zu reduzieren. Außerdem wird das Resharding schneller abgeschlossen, da bei der Slot-Verteilung mehr Ressourcen verwendet werden können.
- Überprüfen Sie das Client-Timeout-Verhalten – Einige Clients stellen möglicherweise eine höhere Latenz während der Online-Cluster-Größenänderung fest. Es kann helfen, bei Ihrer Client-Bibliothek einen höheren Timeout zu konfigurieren, da dem System so Zeit zur Verbindungsherstellung unter höheren Lastbedingungen auf dem Server gegeben wird. Manchmal wird eine große Anzahl an Verbindungen zum Server geöffnet. Fügen Sie in diesen Fällen exponentielles Backoff hinzu, um Logik erneut zu verbinden. Hierdurch wird verhindert, dass ein Schub neuer Verbindungen den Server gleichzeitig erreicht.
- Laden Sie Ihre Funktionen auf jeden Shard — Beim Skalieren Ihres Clusters ElastiCache werden die Funktionen, die in einem der vorhandenen Knoten geladen wurden (zufällig ausgewählt), automatisch auf die neuen Knoten repliziert. Wenn Ihr Cluster über Valkey 7.2 und höher oder Redis OSS 7.0 oder höher verfügt und Ihre Anwendung [Functions](#) verwendet, empfehlen wir, alle Ihre Funktionen vor dem Skalieren auf alle Shards zu laden, damit Ihr Cluster nicht mit unterschiedlichen Funktionen auf verschiedenen Shards endet.

Beachten Sie nach dem Resharding Folgendes:

- Die Skalierung nach oben ist möglicherweise nur zum Teil erfolgreich, wenn auf den Ziel-Shards nicht ausreichend Arbeitsspeicher verfügbar ist. In diesem Fall prüfen Sie den verfügbaren Speicher und wiederholen Sie ggf. die Operation. Die Daten auf den Ziel-Shards werden nicht gelöscht.
- Slots mit großen Elementen werden nicht migriert. Dies gilt besonders für Slots mit Elementen, die nach der Serialisierung größer als 256 MB sind.
- Die Befehle FLUSHALL und FLUSHDB werden in Lua-Skripten während eines Resharding-Vorgangs nicht unterstützt. Vor Redis OSS 6 wurde der BRPOPLPUSH Befehl nicht unterstützt, wenn er auf dem zu migrierenden Slot ausgeführt wird.

## Minimieren der Ausfallzeit während der Wartung

Die Konfiguration des Cluster-Modus ist am besten im Rahmen verwalteter oder nicht verwalteter Operationen verfügbar. Es wird empfohlen, einen im Cluster-Modus unterstützten Client zu verwenden, der eine Verbindung mit dem Clusterermittlungsendpunkt herstellt. Bei deaktiviertem Cluster-Modus empfehlen wir, den primären Endpunkt für alle Schreiboperationen zu verwenden.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren. Bei deaktiviertem Cluster-Modus empfehlen wir daher, den Reader-Endpunkt für Leseaktivitäten zu verwenden.

Wenn im Cluster aktiviert AutoFailover ist, kann sich der primäre Knoten ändern. Daher sollte die Anwendung die Rolle des Knotens bestätigen und alle Leseendpunkte aktualisieren. Dadurch wird sichergestellt, dass Sie keine große Belastung des primären Knotens verursachen. Bei AutoFailover deaktivierter Option ändert sich die Rolle des Knotens nicht. Die Ausfallzeiten bei verwalteten oder nicht verwalteten Vorgängen sind jedoch höher als bei Clustern mit AutoFailover aktivierter Option.

Vermeiden Sie es, Leseanforderungen an einen einzelnen Read-Replica-Knoten weiterzuleiten, da dessen Nichtverfügbarkeit zu einem Leseausfall führen könnte. Entweder Sie greifen auf das Lesen vom Primärknoten zurück oder stellen sicher, dass Sie über mindestens zwei Read Replicas verfügen, um Leseunterbrechungen während der Wartung zu vermeiden.

# Caching-Strategien für Memcached

Im folgenden Thema finden Sie Strategien zum Auffüllen und Verwalten Ihres Memcached-Caches.

Welche Strategien Sie zum Auffüllen und Verwalten Ihres Cache implementieren müssen, hängt von den zwischengespeicherten Daten und den Zugriffsmustern auf diese Daten ab. Zum Beispiel möchten Sie wahrscheinlich nicht dieselbe Strategie für eine Top-10-Bestenliste auf einer Spieleseite und für trendige Nachrichten verwenden. Im Rest dieses Abschnitts besprechen wir gängige Cache-Wartungsstrategien und ihre Vor- und Nachteile.

## Themen

- [Lazy Loading](#)
- [Write-Through](#)
- [Hinzufügen TTL](#)
- [Verwandte Themen](#)

## Lazy Loading

Wie der Name schon sagt, ist Lazy Loading ("langsames Laden") eine Caching-Strategie, die Daten nur bei Bedarf in den Cache lädt. Es funktioniert folgendermaßen.

Amazon ElastiCache ist ein speicherinterner Schlüsselwertspeicher, der sich zwischen Ihrer Anwendung und dem Datenspeicher (Datenbank) befindet, auf den sie zugreift. Immer wenn Ihre Anwendung Daten anfordert, sendet sie die Anfrage zuerst an den Cache. ElastiCache Wenn die Daten im Cache vorhanden und aktuell sind, werden die Daten an Ihre Anwendung ElastiCache zurückgegeben. Wenn die Daten nicht im Cache vorhanden sind oder abgelaufen sind, fordert Ihre Anwendung die Daten von Ihrem Datenspeicher an. Ihr Datenspeicher gibt die Daten dann an Ihre Anwendung zurück. Als nächstes schreibt Ihre Anwendung die vom Speicher empfangenen Daten in den Cache. Auf diese Weise können sie bei der nächsten Anforderung schneller abgerufen werden.

Ein Cache-Treffer tritt auf, wenn sich Daten im Cache befinden und nicht abgelaufen sind:

1. Die Anwendung fordert Daten aus dem Cache an.
2. Der Cache gibt die Daten an die Anwendung zurück.

Ein Cache-Fehlertreffer tritt auf, wenn sich Daten nicht im Cache befinden oder abgelaufen sind:

1. Die Anwendung fordert Daten vom Cache an.
2. Der Cache enthält die angeforderten Daten nicht und gibt daher null zurück.
3. Ihre Anwendung fordert die Daten an und erhält sie aus der Datenbank.
4. Ihre Anwendung aktualisiert den Cache mit den neuen Daten.

## Vor- und Nachteile von Lazy Loading

Die Vorteile von Lazy Loading sind:

- Es werden nur die angeforderten Daten im Cache abgelegt.

Da die meisten Daten nie angefordert werden, wird durch Lazy Loading vermieden, dass der Cache mit nicht angeforderten Daten gefüllt wird.

- Knotenfehler sind für Ihre Anwendung nicht fatal.

Wenn ein Knoten ausfällt und durch einen neuen, leeren Knoten ersetzt wird, funktioniert Ihre Anwendung weiterhin, allerdings mit erhöhter Latenz. Wenn Anforderungen an den neuen Knoten gestellt werden, führt jeder Cache-Fehlertreffer zu einer Abfrage der Datenbank. Gleichzeitig wird die Datenkopie dem Cache hinzugefügt, so dass nachfolgende Anforderungen aus dem Cache abgerufen werden.

Die Nachteile von Lazy Loading sind:

- Bei Cache-Fehlschlägen gibt es Verzögerungen. Jeder Cache-Fehlertreffer führt zu drei Übertragungsvorgängen:

1. Anfängliche Anforderung von Daten aus dem Cache
2. Abfrage der Daten aus der Datenbank
3. Schreiben der Daten in den Cache

Diese Fehlertreffer können zu einer merklichen Verzögerung beim Eingang der Daten in die Anwendung führen.

- Veraltete Daten.

Wenn Daten nur bei einem Cache-Fehlertreffer in den Cache geschrieben werden, können Daten im Cache veraltet sein. Dieses Ergebnis tritt auf, weil der Cache nicht aktualisiert wird, wenn Daten in der Datenbank geändert werden. Um dieses Problem zu beheben, können Sie die [Write-Through](#) und [Hinzufügen TTL](#)-Strategien anwenden.

## Beispiel für Lazy Loading Pseudocode

Im Folgenden finden Sie ein Pseudocode-Beispiel für Lazy Loading-Logik.

```
// *****
// 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
```

In diesem Beispiel sieht der Anwendungscode, der die Daten abrufen, wie folgt aus.

```
customer_record = get_customer(12345)
```

## Write-Through

Die Write-Through-Strategie fügt dem Cache Daten hinzu oder aktualisiert Daten im Cache, wenn Daten in die Datenbank geschrieben werden.

### Vor- und Nachteile von Write-Through

Die Vorteile von Write-Through sind:

- Die Daten im Cache sind nie veraltet.

Da die Daten im Cache jedes Mal aktualisiert werden, wenn sie in die Datenbank geschrieben werden, sind die Daten im Cache immer aktuell.

- Schreibstrafe im Vergleich zu Lesestrafe.

Jeder Schreibvorgang umfasst zwei Übertragungsvorgänge:

1. Ein Schreibvorgang in den Cache
2. Ein Schreibvorgang in die Datenbank

Dadurch wird die Latenz des Prozesses erhöht. Endbenutzer tolerieren im Allgemeinen Latenz beim Aktualisieren von Daten eher als beim Abrufen. Aktualisierungen werden als arbeits- und zeitintensiver wahrgenommen.

Die Nachteile von Write-Through sind:

- Fehlende Daten.

Wenn Sie einen neuen Knoten hochfahren, sei es aufgrund eines Knotenfehlers oder einer horizontalen Skalierung, fehlen Daten. Diese Daten fehlen weiterhin, bis sie in der Datenbank hinzugefügt oder aktualisiert werden. Sie können dies minimieren, indem Sie [Lazy Loading](#) mit Write-Through implementieren.

- Cache-Änderung.

Die meisten Daten werden nie gelesen, was eine Verschwendung von Ressourcen darstellt. Durch [Hinzufügen eines Time-to-Live \(TTL\) -Werts](#) können Sie verschwendeten Speicherplatz minimieren.

Beispiel für Write-Through-Pseudocode

Das Folgende ist ein Pseudocode-Beispiel für Write-Through-Logik.

```
// *****
// 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
```

In diesem Beispiel sieht der Anwendungscode, der die Daten abrufen, wie folgt aus.

```
save_customer(12345, {"address": "123 Main"})
```

## Hinzufügen TTL

Lazy Loading ermöglicht veraltete Daten, schlägt jedoch nicht wegen leeren Knoten fehl. Write-Through stellt sicher, dass die Daten immer frisch sind, kann aber bei leeren Knoten fehlschlagen und den Cache mit überflüssigen Daten füllen. Indem Sie jedem Schreibvorgang einen Wert für time to live (TTL) hinzufügen, können Sie die Vorteile der einzelnen Strategien nutzen. Gleichzeitig können Sie das Überladen des Caches mit zusätzlichen Daten weitgehend vermeiden.

Time to live (TTL) ist ein ganzzahliger Wert, der die Anzahl der Sekunden angibt, bis der Schlüssel abläuft. Valkey oder Redis OSS können Sekunden oder Millisekunden für diesen Wert angeben. Memcached gibt diesen Wert in Sekunden an. Wenn eine Anwendung versucht, einen abgelaufenen Schlüssel zu lesen, gilt der Schlüssel als nicht gefunden. Die Datenbank wird nach dem Schlüssel abgefragt und der Cache aktualisiert. Dieser Ansatz garantiert nicht, dass ein Wert nicht veraltet ist. Er verhindert jedoch, dass die Daten zu veraltet werden, und erfordert, dass Werte im Cache gelegentlich aus der Datenbank aktualisiert werden.

[Weitere Informationen finden Sie unter den Befehlen Valkey und OSS Redis oder den Memcached-Befehlen. set](#)

### TTLBeispiele für Pseudocode

Im Folgenden finden Sie ein Pseudocode-Beispiel für Write-Through-Logik mit 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
```

Das Folgende ist ein Pseudocode-Beispiel für Lazy-Loading-Logik mit 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
```

In diesem Beispiel sieht der Anwendungscode, der die Daten abrufen, wie folgt aus.

```
save_customer(12345, {"address": "123 Main"})
```

```
customer_record = get_customer(12345)
```

## Verwandte Themen

- [In-Memory-Datastore](#)
- [Auswählen einer Engine und einer Version](#)
- [Skalierung ElastiCache](#)

## Verwaltung Ihres selbst entworfenen Clusters in ElastiCache

ElastiCache bietet zwei Bereitstellungsoptionen: serverloses Caching und selbst entworfene Cluster. Jeder hat seine eigenen Fähigkeiten und Anforderungen.

Dieser Abschnitt enthält Themen, die Ihnen bei der Verwaltung Ihrer selbst entworfenen Cluster helfen sollen.

 Note

Diese Themen gelten nicht für ElastiCache Serverless.

## Themen

- [Auto Scaling Valkey- und Redis-Cluster OSS](#)
- [Ändern des Clustermodus](#)
- [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#)
- [Hohe Verfügbarkeit mit Replikationsgruppen](#)
- [Verwaltung der ElastiCache Cluster-Wartung](#)
- [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#)

## Auto Scaling Valkey- und Redis-Cluster OSS

### Voraussetzungen

ElastiCache Auto Scaling ist auf Folgendes beschränkt:

- Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert), auf denen Valkey 7.2 oder Redis OSS Engine Version 6.0 oder höher ausgeführt wird
- Cluster mit Datenklassierung (Clustermodus aktiviert), auf denen Valkey 7.2 oder Redis Engine ab Version 7.0.7 ausgeführt wird OSS
- Instanzgrößen: Groß, 2 XLarge XLarge
- Instance-Familien – R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- Auto Scaling in ElastiCache wird nicht für Cluster unterstützt, die in globalen Datenspeichern, Outposts oder Local Zones ausgeführt werden.

## Automatisches Kapazitätsmanagement mit ElastiCache Auto Scaling mit Valkey oder Redis OSS

ElastiCache Auto Scaling mit Valkey oder Redis OSS ist die Möglichkeit, die gewünschten Shards oder Repliken in Ihrem Service automatisch zu erhöhen oder zu verringern. ElastiCache ElastiCache nutzt den Application Auto Scaling Scaling-Dienst, um diese Funktionalität bereitzustellen. Weitere Informationen finden Sie unter [Application Auto Scaling](#). Um die automatische Skalierung zu verwenden, definieren und wenden Sie eine Skalierungsrichtlinie an, die von Ihnen CloudWatch zugewiesene Metriken und Zielwerte verwendet. ElastiCache Auto Scaling verwendet die Richtlinie, um die Anzahl der Instanzen als Reaktion auf tatsächliche Workloads zu erhöhen oder zu verringern.

Sie können die verwenden AWS Management Console , um eine Skalierungsrichtlinie anzuwenden, die auf einer vordefinierten Metrik basiert. Eine `predefined metric` ist in einer Aufzählung definiert, sodass Sie sie im Code durch einen Namen angeben oder in der AWS Management Console verwenden können. Benutzerdefinierte Metriken können nicht über die AWS Management Console ausgewählt werden. Alternativ können Sie entweder das AWS CLI oder das Application Auto Scaling verwenden, API um eine Skalierungsrichtlinie anzuwenden, die auf einer vordefinierten oder benutzerdefinierten Metrik basiert.

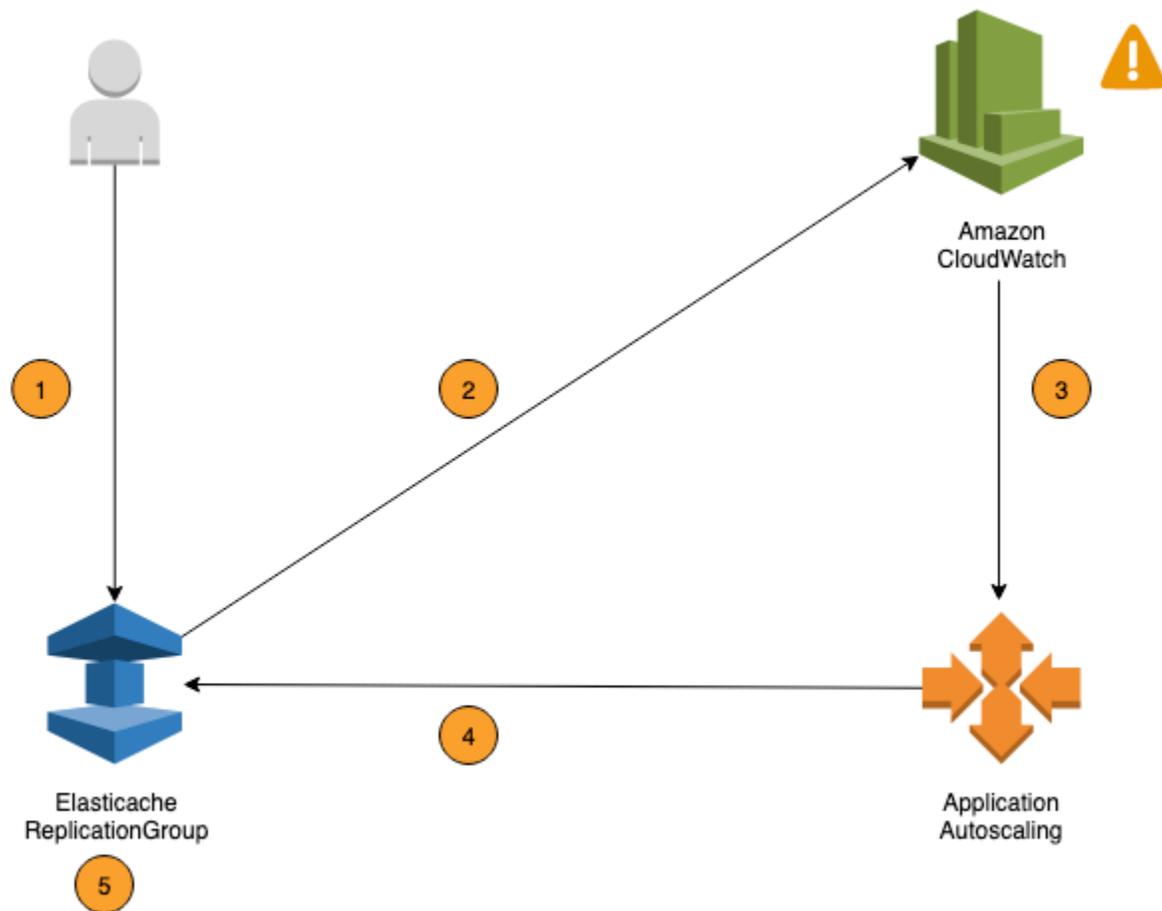
ElastiCache mit Valkey oder Redis wird OSS die Skalierung für die folgenden Dimensionen unterstützt:

- **Shards**— Automatisches Hinzufügen/Entfernen von Shards im Cluster ähnlich wie beim manuellen Online-Resharding. In diesem Fall löst ElastiCache Auto Scaling die Skalierung in Ihrem Namen aus.
- **Replicas**— Automatisches Hinzufügen/Entfernen von Replikaten im Cluster ähnlich wie bei manuellen Erhöhen/Verringern von Replikaten. ElastiCache mit Valkey oder Redis OSS Auto Scaling werden Replikate einheitlich für alle Shards im Cluster hinzugefügt/entfernt.

ElastiCache mit Valkey oder Redis werden die folgenden Arten von Richtlinien für die automatische Skalierung unterstütztOSS:

- [Skalierungsrichtlinien für die Ziel-Nachverfolgung](#) Erhöhen oder Verringern der Anzahl der Aufgaben, die von Ihrem Service ausgeführt werden, auf Grundlage eines Zielwerts für eine bestimmte Metrik. Dies ähnelt der Art und Weise, wie ein Thermostat die Temperatur in Ihrem Zuhause konstant hält. Sie wählen eine Temperatur aus und der Thermostat erledigt den Rest.

- [Geplante Skalierung für Anwendungen ElastiCache mit Valkey oder Redis OSS Auto Scaling](#) — Erhöhen oder verringern Sie die Anzahl der Shard/Replikate, die Ihr Service ausführt, je nach Datum und Uhrzeit.



Die folgenden Schritte fassen den OSS auto Skalierungsprozess ElastiCache mit Valkey oder Redis zusammen, wie im vorherigen Diagramm dargestellt:

1. Sie erstellen eine ElastiCache Auto Scaling-Richtlinie für Ihre Replikationsgruppe.
2. ElastiCache Die auto Skalierung mit Valkey oder Redis OSS erstellt in Ihrem Namen ein Paar CloudWatch Alarme. Jedes Paar stellt die Ober- und Untergrenze für Metriken dar. Diese CloudWatch Alarme werden ausgelöst, wenn die tatsächliche Auslastung des Clusters über einen längeren Zeitraum von Ihrer Zielauslastung abweicht. Sie können jetzt -Alarme in der -Konsole anzeigen.

3. Wenn der konfigurierte Metrikwert Ihre Zielauslastung für einen bestimmten Zeitraum überschreitet (oder unter das Ziel fällt), wird ein Alarm CloudWatch ausgelöst, der Auto Scaling zur Bewertung Ihrer Skalierungsrichtlinie auslöst.
4. ElastiCache bei Valkey oder Redis gibt OSS Auto Scaling eine Modifizierungsanforderung aus, um Ihre Clusterkapazität anzupassen.
5. ElastiCache mit Valkey oder Redis wird die Modifizierungsanforderung OSS verarbeitet und dabei die Kapazität der Cluster-Shards/Replicas dynamisch erhöht (oder verringert), sodass sie sich Ihrer Zielauslastung annähert.

Um zu verstehen, wie OSS Auto Scaling ElastiCache mit Valkey oder Redis funktioniert, nehmen wir an, Sie haben einen Cluster mit dem Namen `UsersCluster`. Indem Sie die CloudWatch Metriken überwachen `UsersCluster`, bestimmen Sie die maximale Anzahl an Shards, die der Cluster benötigt, wenn der Verkehr seinen Höhepunkt erreicht hat, und die Mindestanzahl an Shards, wenn der Verkehr am niedrigsten Punkt ist. Sie legen auch einen Zielwert für die CPU Auslastung des `UsersCluster` Clusters fest. ElastiCache Auto Scaling verwendet seinen Target-Tracking-Algorithmus, um sicherzustellen, dass die bereitgestellten Shards von `UsersCluster` nach Bedarf angepasst werden, sodass die Auslastung auf oder nahe dem Zielwert bleibt.

#### Note

Die Skalierung kann viel Zeit in Anspruch nehmen und erfordert zusätzliche Cluster-Ressourcen, damit die Shards wieder ausgeglichen werden können. ElastiCache mit Valkey oder Redis ändert OSS Auto Scaling die Ressourceneinstellungen nur, wenn die tatsächliche Arbeitslast über einen längeren Zeitraum von mehreren Minuten erhöht (oder reduziert) bleibt. Der Auto Scaling Target-Tracking-Algorithmus versucht, die Zielauslastung langfristig auf oder nahe dem von Ihnen gewählten Wert zu halten.

## Auto-Scaling-Richtlinien

Eine Skalierungsrichtlinie enthält die folgenden Komponenten:

- Eine Zielmetrik — Die CloudWatch Metrik, die ElastiCache bei Valkey oder Redis OSS Auto Scaling verwendet wird, um zu bestimmen, wann und wie stark skaliert werden soll.
- Minimale und maximale Kapazität — Die minimale und die maximale Anzahl an Shards oder Replikaten, die für die Skalierung verwendet werden sollen.

**⚠ Important**

Wenn bei der Erstellung der Auto Scaling-Richtlinie die aktuelle Kapazität höher als die konfigurierte maximale Kapazität ist, gehen wir `scaleIn MaxCapacity` bei der Erstellung der Richtlinie vor. Wenn die aktuelle Kapazität niedriger als die konfigurierte Mindestkapazität ist, gehen wir `scaleOut` in ähnlicher Weise vor `MinCapacity`.

- Eine Cooldown-Phase: Die Zeitspanne (in Sekunden) zwischen dem Ende einer Scale-In- bzw. einer Scale-Out-Aktivität und dem Beginn einer neuen Scale-Out-Aktivität.
- Eine dienstverknüpfte Rolle — Eine AWS Identity and Access Management Zugriffsverwaltungsrolle (IAM), die mit einem bestimmten AWS Dienst verknüpft ist. Eine dienstbezogene Rolle umfasst alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen. ElastiCache mit Valkey oder Redis OSS Auto Scaling generiert diese Rolle automatisch für Sie. `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`
- Scale-In-Aktivitäten aktivieren oder deaktivieren: Die Möglichkeit, Scale-In-Aktivitäten für eine Richtlinie zu aktivieren oder zu deaktivieren.

## Themen

- [Zielmetrik für Auto Scaling](#)
- [Minimale und maximale Kapazität](#)
- [Abkühlungszeit](#)
- [Aktivieren oder Deaktivieren von Scale-In-Aktivitäten](#)

## Zielmetrik für Auto Scaling

Bei dieser Art von Richtlinie werden eine vordefinierte oder benutzerdefinierte Metrik und ein Zielwert für die Metrik in einer Konfiguration für die Skalierungsrichtlinie zur Zielverfolgung angegeben. ElastiCache mit Valkey oder Redis erstellt und verwaltet OSS Auto Scaling CloudWatch Alarme, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung auf der Grundlage der Metrik und des Zielwerts. Die Skalierungsrichtlinie fügt -Repliken hinzu oder entfernt sie, wenn erforderlich, um die Metrik auf dem angegebenen Zielwert oder in der Nähe davon zu halten. Abgesehen davon, dass eine Skalierungsrichtlinie für die Ziel-Nachverfolgung die Metrik nahe an dem Zielwert hält, passt sie sich auch an die Schwankungen in der Metrik aufgrund eines sich

ändernden Workloads an. Eine solche Richtlinie minimiert auch schnelle Schwankungen in der Anzahl der verfügbaren -Repliken für Ihren DB-Cluster.

Nehmen wir zum Beispiel eine Skalierungsrichtlinie, die die vordefinierte durchschnittliche `ElastiCachePrimaryEngineCPUUtilization`-Metrik verwendet. Mit einer solchen Richtlinie kann die CPU Auslastung auf oder nahe einem bestimmten Nutzungsprozentsatz, z. B. 70 Prozent, gehalten werden.

#### Note

Für jeden DB-Cluster können Sie nur eine Auto-Scaling-Richtlinie für jede Zielmetrik erstellen.

## Minimale und maximale Kapazität

### Shards

Sie können die maximale Anzahl von Shards angeben, auf die ElastiCache mit Valkey oder OSS Redis Auto Scaling skaliert werden kann. Dieser Wert muss kleiner oder gleich 250 sein und mindestens 1 sein. Sie können auch die Mindestanzahl von Shards angeben, die mit Auto Scaling verwaltet werden sollen. Dieser Wert muss mindestens 1 betragen und gleich oder kleiner sein als der Wert, der für die maximale Anzahl von Scherben 250 angegeben ist.

### Replikas

Sie können die maximale Anzahl von Replikaten angeben, die ElastiCache mit Valkey oder Redis OSS Auto Scaling verwaltet werden sollen. Dieser Wert muss gleich oder kleiner 5 sein. Sie können auch die Mindestanzahl von Replikaten angeben, die mit Auto Scaling verwaltet werden sollen. Dieser Wert muss mindestens 1 betragen und gleich oder kleiner als der für die maximale Anzahl von Replikaten angegebene Wert 5 sein.

Um die minimale und maximale Anzahl von Shards / Replikaten zu ermitteln, die Sie für einen typischen Datenverkehr benötigen, testen Sie Ihre Auto-Scaling-Konfiguration mit der erwarteten Datenverkehrsrate für Ihr Modell.

#### Note

ElastiCache erhöhen Sie mit den OSS Auto Scaling-Richtlinien von Valkey oder Redis die Clusterkapazität, bis die von Ihnen definierte Maximalgröße erreicht ist oder bis

Servicebeschränkungen gelten. Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

### Important

Scaling-in findet statt, wenn es keinen Verkehr gibt. Wenn der Traffic einer Variante Null wird, ElastiCache skaliert Valkey oder Redis OSS automatisch auf die angegebene Mindestanzahl von Instanzen.

## Abkühlungszeit

Sie können die Reaktionsfähigkeit einer Skalierungsrichtlinie für die Zielverfolgung anpassen, indem Sie Ruhephasen hinzufügen, nachdem eine Skalierung Ihres DB-Clusters erfolgt ist. Eine Ruhephase blockiert nachfolgende Scale-in- oder Scale-out-Anforderungen bis zum Ablauf der Frist. Dies verlangsamt das Löschen von Shard/Replikaten in Ihrem OSS Cluster ElastiCache mit Valkey oder Redis für Scale-In-Anfragen und die Erstellung von Shard/Replikaten für Scale-Out-Anfragen. Sie können die folgenden Ruhephasen angeben:

- Eine Scale-In-Aktivität reduziert die Anzahl der Shard/Repliken in Ihrem Cluster. Eine Ruhephase zum Abskalieren gibt die Zeitspanne in Sekunden an, nach der eine Abskalierung abgeschlossen sein muss, bevor eine weitere Abskalierung gestartet werden kann.
- Eine Scale-Out-Aktivität erhöht die Anzahl der Shard/Replikate in Ihrem Cluster. Eine Ruhephase zum Aufskalieren gibt die Zeitspanne in Sekunden an, nach der eine Aufskalierung abgeschlossen sein muss, bevor eine weitere Aufskalierung gestartet werden kann.

Wenn keine Ruhephase für das Abskalieren oder Aufskalieren angegeben ist, beträgt der Standardwert für die Aufskalierung 600 Sekunden und für das Abskalieren 900 Sekunden.

## Aktivieren oder Deaktivieren von Scale-In-Aktivitäten

Sie können die Scale-In-Aktivitäten für eine Richtlinie aktivieren oder deaktivieren. Die Aktivierung von Scale-In-Aktivitäten ermöglicht es der Skalierungsrichtlinie, -Repliken zu löschen. Wenn die Scale-In-Aktivitäten aktiviert sind, gilt für die Scale-In-Aktivitäten die Scale-In-Ruhezeit in der Skalierungsrichtlinie. Das Deaktivieren von Scale-In-Aktivitäten verhindert das Löschen von -Repliken durch die Skalierungsrichtlinie.

**Note**

Scale-Out-Aktivitäten sind immer aktiviert, sodass die Skalierungsrichtlinie bei Bedarf Shard/Replicas mit Valkey oder Redis erstellen ElastiCache kann. OSS

## IAM Für Auto Scaling erforderliche Berechtigungen

ElastiCache mit Valkey oder Redis wird OSS Auto Scaling durch eine Kombination aus ElastiCache CloudWatch, und Application Auto Scaling ermöglicht. APIs Cluster werden mit ElastiCache (RedisOSS) erstellt und aktualisiert, Alarme werden mit CloudWatch Application Auto Scaling erstellt und Skalierungsrichtlinien werden erstellt. Zusätzlich zu den IAM Standardberechtigungen für das Erstellen und Aktualisieren von Clustern muss der IAM Benutzer, der auf die ElastiCache Auto Scaling Scaling-Einstellungen zugreift, über die entsprechenden Berechtigungen für die Dienste verfügen, die dynamische Skalierung unterstützen. IAM Benutzer müssen über Berechtigungen verfügen, um die in der folgenden Beispielrichtlinie aufgeführten Aktionen verwenden zu können:

```
{
 "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"
}
]
```

## Servicegebundene Rolle

Der OSS Auto Scaling-Dienst ElastiCache with Valkey oder Redis benötigt außerdem die Erlaubnis, Ihre Cluster und CloudWatch Alarme zu beschreiben, sowie Berechtigungen, Ihre ElastiCache Zielkapazität in Ihrem Namen zu ändern. Wenn Sie Auto Scaling für Ihren Cluster aktivieren, wird eine dienstverknüpfte Rolle mit dem Namen `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG` erstellt. Diese dienstbezogene Rolle gewährt ElastiCache Auto Scaling-Berechtigungen, um die Alarme für Ihre Richtlinien zu beschreiben, die aktuelle Kapazität der Flotte zu überwachen und die Kapazität der Flotte zu ändern. Die serviceverknüpfte Rolle ist die Standardrolle für ElastiCache Auto Scaling. Weitere Informationen finden Sie unter [Dienstbezogene Rollen für ElastiCache \(RedisOSS\) Auto Scaling im Application Auto Scaling](#) Scaling-Benutzerhandbuch.

## Bewährte Methoden für die Auto Scaling

Wir empfehlen vor der Registrierung für Auto Scaling Folgendes:

1. Verwenden Sie nur eine Tracking-Metrik — Identifizieren Sie, ob Ihr Cluster datenintensive Workloads hat CPU, und verwenden Sie eine entsprechende vordefinierte Metrik, um die Skalierungsrichtlinie zu definieren.
  - EngineCPU: `ElastiCachePrimaryEngineCPUUtilization` (Shard-Dimension) oder `ElastiCacheReplicaEngineCPUUtilization` (Replikat-Dimension)
  - Datenbanknutzung: `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` Diese Skalierungsrichtlinie funktioniert am besten, wenn „maxmemory-policy“ im Cluster auf „noeviction“ festgelegt ist.

Wir empfehlen, mehrere Richtlinien pro Dimension auf dem Cluster zu vermeiden. ElastiCache Mit Valkey oder Redis skaliert OSS Auto Scaling das skalierbare Ziel, wenn Richtlinien für die Zielverfolgung bereit sind, aber es wird nur dann skaliert, wenn alle Zielverfolgungsrichtlinien (mit

aktiviertem Scale-In-Teil) skalierbar sind. Wenn mehrere Richtlinien das skalierbare Ziel anweisen, gleichzeitig zu herauf oder herunter zu skalieren, skaliert auf Grundlage der Richtlinie, die die größte Kapazität für das Herauf- und Herunterskalieren bietet.

2. Benutzerdefinierte Metriken für die Zielverfolgung – Seien Sie vorsichtig, wenn Sie benutzerdefinierte Metriken für die Zielverfolgung verwenden, da die automatische Skalierung am besten geeignet ist, um proportional zu Änderungen der für die Richtlinie ausgewählten Metriken auf-/abzuskalieren. Wenn solche Metriken, die sich nicht proportional zu den Skalierungsaktionen ändern, zur Richtlinienerstellung verwendet werden, kann dies zu kontinuierlichen Auf- oder Abskalierungsaktionen führen, was sich auf Verfügbarkeit oder Kosten auswirken kann.

Vermeiden Sie bei Daten-Tiering-Clustern (Instance-Familie r6gd) die Verwendung speicherbasierter Metriken für die Skalierung.

3. Geplante Skalierung – Wenn Sie feststellen, dass Ihre Workload deterministisch ist (erreichen Sie hoch/niedrig zu einem bestimmten Zeitpunkt), empfehlen wir Ihnen, die geplante Skalierung zu verwenden und Ihre Zielkapazität entsprechend den Anforderungen zu konfigurieren. Target Tracking eignet sich am besten für nicht-deterministische Workloads und für einen Cluster für den Betrieb mit der erforderlichen Zielmetrik, indem er aufskaliert, wenn Sie mehr Ressourcen benötigen, und abskaliert, wenn Sie weniger benötigen.
4. Deaktivieren von Abskalierungen – Die automatische Skalierung mit Target Tracking eignet sich am besten für Cluster mit allmählicher Zunahme/Abnahme des Workloads, da Spitzen/Einbrüche in den Metriken aufeinanderfolgende Schwankungen bei Auf-/Abskalierungen auslösen können. Um solche Schwankungen zu vermeiden, können Sie mit deaktivierter Abskalierung beginnen. Später können Sie jederzeit manuell nach Ihren Bedürfnissen abskalieren.
5. Testen Ihrer Anwendung – Wir empfehlen Ihnen, Ihre Anwendung mit Ihren geschätzten minimalen und maximalen Workloads zu testen, um die absoluten minimalen und maximalen Shards/ Replikate zu ermitteln, die für den Cluster erforderlich sind, während Sie Skalierungsrichtlinien erstellen, um Verfügbarkeitsprobleme zu vermeiden. Die automatische Skalierung kann bis zum Maximal- und Minimalwert auf- bzw. abskalieren, der für das Ziel konfiguriert wurde.
6. Definition des Zielwerts — Sie können die entsprechenden CloudWatch Metriken für die Clusterauslastung über einen Zeitraum von vier Wochen analysieren, um den Schwellenwert für den Zielwert zu ermitteln. Wenn Sie sich immer noch nicht sicher sind, welchen Wert Sie wählen möchten, empfehlen wir, mit dem minimal unterstützten vordefinierten Metrikwert zu beginnen.
7. AutoScaling On Target Tracking eignet sich am besten für Cluster mit einheitlicher Verteilung der Workloads auf die Shard-/Replica-Dimension. Eine ungleichmäßige Verteilung kann zu folgenden Faktoren führen:

- Skalierung, wenn nicht erforderlich, aufgrund von Workload Spike/Dip auf ein paar Hot-Shards/Repliken.
- Keine Skalierung, wenn erforderlich, da insgesamt durchschnittlich nahe am Ziel liegt, obwohl Hot-Shards/Replikate vorhanden sind.

#### Note

Bei der Skalierung Ihres Clusters ElastiCache werden die Funktionen, die in einem der vorhandenen Knoten geladen wurden (zufällig ausgewählt), automatisch auf die neuen Knoten repliziert. Wenn Ihr Cluster über Valkey oder Redis OSS 7.0 oder höher verfügt und Ihre Anwendung [Functions](#) verwendet, empfehlen wir, alle Ihre Funktionen vor dem Skalieren auf alle Shards zu laden, damit Ihr Cluster nicht mit unterschiedlichen Funktionen auf verschiedenen Shards endet.

Beachten Sie nach der Registrierung Folgendes AutoScaling:

- Es gibt Einschränkungen bei den unterstützten Konfigurationen für die automatische Skalierung, daher empfehlen wir, die Konfiguration einer Replikationsgruppe, die für die automatische Skalierung registriert ist, nicht zu ändern. Im Folgenden sind einige Beispiele aufgeführt:
  - Manuelles Ändern des Instance-Typs in nicht unterstützte Typen.
  - Zuordnen der Replikationsgruppe zu einem globalen Datenspeicher.
  - Ändern `ReservedMemoryPercent`-Parameter.
  - Manuelles Erhöhen/Verringern von Shards/Replikaten über die bei der Richtlinienerstellung konfigurierte Mindest- und Maximalkapazität.

## Verwenden von Auto Scaling mit Shards

Damit ElastiCache können AutoScaling Sie Tracking- und geplante Richtlinien mit Ihrer Valkey- oder OSS Redis-Engine verwenden.

Im Folgenden finden Sie Einzelheiten zur Zielverfolgung und zu geplanten Richtlinien sowie deren Anwendung mithilfe von und. AWS Management Console AWS CLI APIs

### Themen

- [Skalierungsrichtlinien für die Ziel-Nachverfolgung](#)

- [Hinzufügen einer Skalierungsrichtlinie](#)
- [Registrieren eines skalierbaren Ziels](#)
- [Definieren einer Skalierungsrichtlinie](#)
- [Deaktivieren der Scale-In-Aktivität](#)
- [Anwenden einer Skalierungsrichtlinie](#)
- [Bearbeiten einer Skalierungsrichtlinie](#)
- [Löschen einer Skalierungsrichtlinie](#)
- [AWS CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden](#)
- [Geplante Skalierung](#)

## Skalierungsrichtlinien für die Ziel-Nachverfolgung

Mit Skalierungsrichtlinien für die Zielverfolgung wählen Sie eine Metrik aus und legen einen Zielwert fest. ElastiCache mit Valkey oder Redis erstellt und verwaltet OSS Auto Scaling die CloudWatch Alarme, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung auf der Grundlage der Metrik und des Zielwerts. Durch die Skalierungsrichtlinie wird so viel Kapazität wie erforderlich hinzugefügt oder entfernt, damit die Metrik auf oder nahe an dem Zielwert gehalten wird. Abgesehen davon, dass eine Skalierungsrichtlinie für die Ziel-Nachverfolgung die Metrik nahe an dem Zielwert hält, passt sie sich auch an die Schwankungen in der Metrik aufgrund eines schwankenden Lastmusters an und verringert schnelle Schwankungen der Kapazität der Flotte.

Nehmen wir zum Beispiel eine Skalierungsrichtlinie, die die vordefinierte `ElastiCachePrimaryEngineCPUUtilization`-Durchschnittsmetrik mit konfigurierbarem Zielwert verwendet. Mit einer solchen Richtlinie kann die CPU Auslastung auf oder nahe dem angegebenen Zielwert gehalten werden.

## Vordefinierte Metriken

Eine vordefinierte Metrik ist eine Struktur, die sich auf einen bestimmten Namen, eine Dimension und eine Statistik (`average`) einer bestimmten CloudWatch Metrik bezieht. Ihre Auto-Scaling-Richtlinie definiert eine der folgenden vordefinierten Metriken für Ihren Cluster:

| Vordefinierter Metrikname                                 | CloudWatch Name der Metrik                     | CloudWatch Metrische Dimension                          | Nicht geeignete Instance-Typen |
|-----------------------------------------------------------|------------------------------------------------|---------------------------------------------------------|--------------------------------|
| ElastiCachePrimaryEngineCPUUtilization                    | EngineCPUUtilization                           | ReplicationGroupId, Rolle = Primär                      | None                           |
| ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage | DatabaseCapacityUsageCountedForEvictPercentage | Metriken für Valkey- oder OSS Redis-Replikationsgruppen | None                           |
| ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage   | DatabaseMemoryUsageCountedForEvictPercentage   | Metriken für Valkey- oder OSS Redis-Replikationsgruppen | R6gd                           |

Instance-Typen mit Datenschicht können nicht verwendet werden `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`, da diese Instance-Typen Daten sowohl im Arbeitsspeicher als auch speichern. SSD Der erwartete Anwendungsfall für datengestützte Instances besteht darin, dass der Speicher zu 100 Prozent genutzt und bei Bedarf aufgefüllt wird. SSD

#### Kriterien für die Auto Scaling für Shards

Wenn der Service erkennt, dass Ihre vordefinierte Metrik gleich oder größer als die Zieleinstellung ist, wird die Shards-Kapazität automatisch erhöht. ElastiCache mit Valkey oder Redis werden Ihre Cluster-Shards um eine Anzahl OSS skaliert, die der größeren von zwei Zahlen entspricht:

prozentuale Abweichung von Target und 20 Prozent der aktuellen Shards. Bei Scale-In erfolgt ElastiCache keine auto Skalierung, es sei denn, der Gesamtwert liegt unter 75 Prozent Ihres definierten Ziels.

Wenn Sie für ein Scal-Out-Beispiel 50 Shards und

- Wenn Ihr Target die Sicherheitslücke um 30 Prozent überschreitet, OSS skaliert es ElastiCache mit Valkey oder Redis um 30 Prozent, was zu 65 Shards pro Cluster führt.
- wenn Ihr Target die Sicherheitslücke um 10 Prozent überschreitet, wird ElastiCache mit Valkey oder Redis standardmäßig mindestens 20 Prozent OSS skaliert, was zu 60 Shards pro Cluster führt.

Ein Beispiel für eine Skalierung: Wenn Sie einen Zielwert von 60 Prozent ausgewählt haben, ElastiCache werden Valkey oder Redis OSS erst auto skalieren, wenn die Metrik 45 Prozent oder weniger beträgt (25 Prozent unter dem Ziel von 60 Prozent).

## Überlegungen zum Auto Scaling

Beachten Sie folgende Überlegungen:

- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung geht davon aus, dass sie eine horizontale Skalierung nach oben vornehmen soll, wenn die angegebene Metrik über dem Zielwert liegt. Sie können keine Skalierungsrichtlinie für die Zielverfolgung verwenden, um eine Skalierung vorzunehmen, wenn die angegebene Metrik unter dem Zielwert liegt. ElastiCache mit Valkey oder Redis werden Shards um eine Abweichung von mindestens 20 Prozent vom Sollwert vorhandener Shards im Cluster OSS skaliert.
- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung nimmt keine Skalierung vor, wenn die angegebene Metrik unzureichende Daten aufweist. Es wird keine Abskalierung vorgenommen, da unzureichende Daten nicht als geringe Auslastung interpretiert werden.
- Möglicherweise werden Lücken zwischen den Datenpunkten für den Zielwert und die aktuelle Metrik angezeigt. Dies liegt daran, dass OSS Auto Scaling ElastiCache bei Valkey oder Redis immer konservativ vorgeht, indem es auf- oder abrundet, wenn es bestimmt, wie viel Kapazität hinzugefügt oder entfernt werden soll. Dadurch wird verhindert, dass zu wenig Kapazität hinzugefügt oder zu viel Kapazität entfernt wird.
- Um die Verfügbarkeit der Anwendung sicherzustellen, wird der Service schnellstmöglich proportional zur Metrik hochskaliert, jedoch etwas langsamer herunterskaliert.

- Sie können mehrere Skalierungsrichtlinien für die Zielverfolgung für einen OSS Cluster ElastiCache mit Valkey oder Redis verwenden, vorausgesetzt, dass jede von ihnen eine andere Metrik verwendet. Die Absicht von ElastiCache (RedisOSS) Auto Scaling besteht darin, der Verfügbarkeit immer Priorität einzuräumen. Daher unterscheidet sich das Verhalten je nachdem, ob die Zielverfolgungsrichtlinien für Scale-Out oder Scale-In bereit sind. Sofern Richtlinien für die Ziel-Nachverfolgung für die Skalierung nach oben bereit sind, findet eine Skalierung des Service nach oben statt. Eine Skalierung nach unten wird jedoch nur vorgenommen, wenn alle Richtlinien für die Ziel-Nachverfolgung (mit aktivierter Skalierung nach unten) zur Skalierung nach unten bereit sind.
- Bearbeiten oder löschen Sie nicht die CloudWatch Alarme, die ElastiCache mit Valkey oder Redis OSS Auto Scaling für eine Skalierungsrichtlinie zur Zielverfolgung verwaltet werden. ElastiCache Auto Scaling löscht die Alarme automatisch, wenn Sie die Skalierungsrichtlinie löschen.
- ElastiCache Auto Scaling verhindert nicht, dass Sie Cluster-Shards manuell ändern. Diese manuellen Anpassungen wirken sich nicht auf bestehende CloudWatch Alarme aus, die mit der Skalierungsrichtlinie verknüpft sind, können sich jedoch auf Metriken auswirken, die diese CloudWatch Alarme auslösen können.
- Diese von Auto Scaling verwalteten CloudWatch Alarme werden über die AVG Metrik für alle Shards im Cluster definiert. So kann Hot-Shards zu einem beliebigen Szenario führen:
  - Skalierung, wenn sie aufgrund der Belastung einiger Hot-Shards, die einen Alarm auslösen, nicht erforderlich ist CloudWatch
  - wird nicht skaliert, wenn dies erforderlich ist, da alle Shards aggregiert werden, AVG was dazu führt, dass der Alarm nicht überschritten wird.
- ElastiCache Bei Valkey oder Redis gelten weiterhin die OSS Standardgrenzwerte für Knoten pro Cluster. Wenn Sie sich also für Auto Scaling entscheiden und erwarten, dass maximale Knoten mehr als die Standardgrenze sind, fordern Sie eine Limiterhöhung bei [AWS -Service-Limits](#) an und wählen Sie den Limit-Typ Knoten pro Cluster pro Instance-Typ.
- Stellen Sie sicher, dass in Ihrem System genügend ENIs (Elastic Network Interfaces) zur Verfügung stehen VPC, die beim Scale-Out benötigt werden. Weitere Informationen finden Sie unter [Elastic-Network-Schnittstellen](#).
- Wenn nicht genügend Kapazität verfügbar ist EC2, wird ElastiCache Auto Scaling nicht skaliert und verzögert, bis die Kapazität verfügbar ist.
- ElastiCache (RedisOSS) Auto Scaling entfernt beim Scale-In keine Shards mit Steckplätzen, die nach der Serialisierung eine Objektgröße von mehr als 256 MB haben.
- Während des Scale-Ins werden Shards nicht entfernt, wenn nicht genügend Speicher für die resultierende Shard-Konfiguration verfügbar ist.

## Hinzufügen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie hinzufügen, indem Sie die verwenden AWS Management Console.

Um eine Auto Scaling Scaling-Richtlinie zu einem Cluster ElastiCache mit Valkey oder OSS Redis hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder OSSRedis aus.
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie Add dynamic scaling (Dynamische Skalierung hinzufügen) aus.
6. Geben Sie unter Policy Name (Richtliniennamen) einen Namen für diese Richtlinie ein.
7. Für Skalierbare Dimension wählen Sie Shards aus.
8. Wählen Sie für die Zielmetrik eine der nachstehenden Optionen aus:
  - Primäre CPU Nutzung, um eine Richtlinie zu erstellen, die auf der durchschnittlichen CPU Auslastung basiert.
  - Memory (Arbeitsspeicher)Um eine Richtlinie zu erstellen, die auf dem durchschnittlichen Datenbankspeicher basiert.
  - Kapazität zum Erstellen einer Richtlinie auf der Grundlage der durchschnittlichen Datenbankkapazitätsauslastung. Die Kapazitätsmetrik umfasst Speicher und SSD Auslastung für datengestützte Instances sowie die Speicherauslastung für alle anderen Instance-Typen.
9. Wählen Sie für den Zielwert einen Wert größer oder gleich 35 und kleiner oder gleich 70 aus. Auto Scaling behält diesen Wert für die ausgewählte Zielmetrik auf allen Ihren ElastiCache Shards bei:
  - Primäre CPU Nutzung: Behält den Zielwert für die EngineCPUUtilization Metrik auf den primären Knoten bei.
  - Speicher: behält den Zielwert für die DatabaseMemoryUsageCountedForEvictPercentage-Metrik bei
  - Kapazität behält den Zielwert für die DatabaseCapacityUsageCountedForEvictPercentage-Metrik bei

Cluster-Shards werden hinzugefügt oder entfernt, um die Metrik in der Nähe des angegebenen Wertes zu halten.

10. (Optional) Die Ruhephase für Scale-In- oder Scale-Out-Ruhephase werden von der Konsole nicht unterstützt. Verwenden Sie die AWS CLI , um die Abklingwerte zu ändern.
11. Geben Sie für Mindestkapazität die Mindestanzahl von Shards ein, für deren Verwaltung die ElastiCache Auto Scaling Scaling-Richtlinie erforderlich ist.
12. Geben Sie für Maximale Kapazität die maximale Anzahl von Shards ein, für deren Verwaltung die ElastiCache Auto Scaling Scaling-Richtlinie erforderlich ist. Dieser Wert muss gleich oder kleiner 250 sein.
13. Wählen Sie Create (Erstellen) aus.

### Registrieren eines skalierbaren Ziels

Bevor Sie Auto Scaling mit einem Cluster ElastiCache mit Valkey oder Redis verwenden können, müssen Sie Ihren OSS Cluster bei ElastiCache Auto Scaling registrieren. Sie tun dies, um die Skalierungsdimension und die Grenzwerte zu definieren, die auf diesen Cluster angewendet werden sollen. ElastiCache Auto Scaling skaliert den Cluster dynamisch entlang der `elasticache:replication-group:NodeGroups` skalierbaren Dimension, die die Anzahl der Cluster-Shards darstellt.

### Unter Verwendung der AWS CLI

Um Ihren OSS Cluster ElastiCache bei Valkey oder Redis zu registrieren, verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern:

- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest
- `--resource-id`— Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters `replication-group/myscalablecluster`.
- `--scalable-dimension` – Legen Sie diesen Wert auf `elasticache:replication-group:NodeGroups` fest.
- `--max-capacity` — Die maximale Anzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden können. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der DB-Instances in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

- `--min-capacity` — Die Mindestanzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden müssen. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der DB-Instances in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

## Example

Im folgenden Beispiel registrieren Sie bei Valkey oder Redis OSS einen Cluster ElastiCache mit dem Namen `myscalablecluster`. Die Registrierung zeigt an, dass der DB-Cluster dynamisch skaliert werden soll, um zwischen einer und acht -Replicas zu haben.

Für Linux, macOS oder 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 \

```

Für 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 ^

```

## Mit dem API

Verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern, um Ihren ElastiCache Cluster zu registrieren:

- `ServiceNamespace` — Setzen Sie diesen Wert auf `elasticache`.
- `ResourceID` — Die Ressourcen-ID für den ElastiCache Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise `replication-group/myscalablecluster` der Name des Clusters.
- `ScalableDimension` — Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.

- **MinCapacity** — Die Mindestanzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden müssen. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).
- **MaxCapacity** — Die maximale Anzahl von Shards, die durch ElastiCache Auto Scaling verwaltet werden können. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

## Example

Im folgenden Beispiel registrieren Sie einen ElastiCache mit Valkey oder Redis benannten OSS Cluster `myscalecluster` mit dem Namen Application Auto Scaling. API Diese Registrierung zeigt an, dass der Cluster dynamisch skaliert werden soll, um zwischen 1 und 5 Replikate zu haben.

```
POST / HTTP/1.1
Host: autoscaling.us-east-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 219
X-Amz-Target: AnyScaleFrontendService.RegisterScalableTarget
X-Amz-Date: 20160506T182145Z
User-Agent: aws-cli/1.10.23 Python/2.7.11 Darwin/15.4.0 botocore/1.4.8
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
 "ServiceNamespace": "elasticache",
 "ResourceId": "replication-group/myscalecluster",
 "ScalableDimension": "elasticache:replication-group:NodeGroups",
 "MinCapacity": 1,
 "MaxCapacity": 5
}
```

## Definieren einer Skalierungsrichtlinie

Eine Konfiguration einer Skalierungsrichtlinie zur Zielverfolgung wird durch einen JSON Block dargestellt, in dem die Metriken und Zielwerte definiert sind. Sie können eine Konfiguration der Skalierungsrichtlinie als JSON Block in einer Textdatei speichern. Sie verwenden diese Textdatei, wenn Sie die AWS CLI oder die Application Auto Scaling API aufrufen. Weitere Informationen zur Syntax der Richtlinienkonfiguration finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der API Referenz zu Application Auto Scaling.

Die folgenden Optionen stehen zur Verfügung, um eine Konfiguration der Skalierungsrichtlinien für die Zielverfolgung zu definieren:

## Themen

- [Verwenden einer vordefinierten Metrik](#)
- [Verwenden einer benutzerdefinierten Metrik](#)
- [Verwenden von Ruhephasen](#)

## Verwenden einer vordefinierten Metrik

Mithilfe vordefinierter Metriken können Sie schnell eine Skalierungsrichtlinie für die Zielverfolgung für einen OSS Cluster ElastiCache mit Valkey oder Redis definieren, der mit Target Tracking in ElastiCache (OSSRedis) Auto Scaling funktioniert.

ElastiCache unterstützt derzeit die folgenden vordefinierten Metriken in NodeGroup Auto Scaling:

- `ElastiCachePrimaryEngineCPUUtilization`— Der Durchschnittswert der `EngineCPUUtilization` Metrik CloudWatch auf allen Primärknoten im Cluster.
- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`— Der Durchschnittswert der `DatabaseMemoryUsageCountedForEvictPercentage` Metrik CloudWatch für alle Primärknoten im Cluster.
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`— Der Durchschnittswert der `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` Metrik CloudWatch für alle Primärknoten im Cluster.

## Weitere Informationen über die Metriken

`EngineCPUUtilization`, `DatabaseMemoryUsageCountedForEvictPercentage` und `DatabaseCapacityUsageCountedForEvictPercentage` Sie unter [Überwachung der Nutzung mit CloudWatch Metrics](#). Um eine vordefinierte Metrik in Ihrer Skalierungsrichtlinie zu verwenden, erstellen Sie eine Zielverfolgungskonfiguration für Ihre Skalierungsrichtlinie. Diese Konfiguration muss einen Wert `PredefinedMetricSpecification` für die vordefinierte Metrik und einen `TargetValue` für den Zielwert dieser Metrik enthalten.

## Example

Das folgende Beispiel beschreibt eine typische Richtlinienkonfiguration für die Skalierung mit Target-Tracking für einen Cluster ElastiCache mit Valkey oder Redis. OSS In dieser Konfiguration wird die `ElastiCachePrimaryEngineCPUUtilization` vordefinierte Metrik verwendet, um den Cluster

auf der Grundlage einer durchschnittlichen CPU Auslastung von 40 Prozent über alle Primärknoten im Cluster anzupassen.

```
{
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 }
}
```

## Verwenden einer benutzerdefinierten Metrik

Durch die Verwendung von benutzerdefinierten Metriken können Sie eine Skalierungsrichtlinie für die Zielverfolgung definieren, die Ihren individuellen Anforderungen entspricht. Sie können eine benutzerdefinierte Metrik auf der Grundlage einer beliebigen ElastiCache Metrik definieren, die sich proportional zur Skalierung ändert. Nicht alle ElastiCache Metriken eignen sich für die Zielverfolgung. Die Metrik muss eine gültige Auslastungsmetrik sein und beschreiben, wie ausgelastet eine Instance ist. Der Wert der Metrik muss sich proportional zur Anzahl der -Replicas im -DB-Cluster erhöhen oder verringern. Diese proportionale Erhöhung oder Verminderung ist notwendig, um die metrischen Daten zur proportionalen Skalierung oder in der Anzahl der -Repliken zu verwenden.

## Example

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt eine benutzerdefinierte Metrik einen ElastiCache (Redis-OSS) Cluster auf der Grundlage einer durchschnittlichen CPU Auslastung von 50 Prozent aller Shards in einem Cluster mit dem Namen `an.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"
 }
}
```

## Verwenden von Ruhephasen

Sie können einen Wert in Sekunden für `ScaleOutCooldown` angeben, um eine Ruhephase für die Aufskalierung Ihres Clusters hinzuzufügen. Ähnlich können Sie einen Wert in Sekunden für `ScaleInCooldown` angeben, um eine Ruhephase für die Abskalierung Ihres Clusters hinzuzufügen. Weitere Informationen finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der API Referenz zu Application Auto Scaling.

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration wird die `ElastiCachePrimaryEngineCPUUtilization` vordefinierte Metrik verwendet, um einen ElastiCache (Redis-OSS) Cluster auf der Grundlage einer durchschnittlichen CPU Auslastung von 40 Prozent über alle Primärknoten in diesem Cluster anzupassen. Die Konfiguration sieht eine Ruhephase von 10 Minuten zum Abskalieren und eine Ruhephase von 5 Minuten zum Aufskalieren vor.

```
{
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 },
 "ScaleInCooldown": 600,
 "ScaleOutCooldown": 300
}
```

## Deaktivieren der Scale-In-Aktivität

Sie können verhindern, dass die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung in Ihrem Cluster skaliert, indem Sie die Scale-In-Aktivität deaktivieren. Das Deaktivieren der Scale-In-Aktivität verhindert das Löschen von -Repliken durch die Skalierungsrichtlinie, und erlaubt der Skalierungsrichtlinie dennoch, Repliken nach Bedarf zu erstellen.

Sie können einen booleschen Wert für `DisableScaleIn` angeben, um die Scale-In-Aktivität für Ihren Cluster zu aktivieren oder zu deaktivieren. Weitere Informationen finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der API Referenz zu Application Auto Scaling.

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt die `ElastiCachePrimaryEngineCPUUtilization` vordefinierte Metrik einen OSS Cluster ElastiCache mit Valkey oder Redis auf der Grundlage einer durchschnittlichen CPU Auslastung von 40 Prozent aller primären Knoten in diesem Cluster an. Die Konfiguration deaktiviert die Scale-In-Aktivität für die Skalierungsrichtlinie.

```
{
 "TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {
 "PredefinedMetricType": "ElastiCachePrimaryEngineCPUUtilization"
 },
 "DisableScaleIn": true
}
```

### Anwenden einer Skalierungsrichtlinie

Nachdem Sie Ihren Cluster ElastiCache bei Valkey oder Redis OSS Auto Scaling registriert und eine Skalierungsrichtlinie definiert haben, wenden Sie die Skalierungsrichtlinie auf den registrierten Cluster an. Um eine Skalierungsrichtlinie auf einen ElastiCache (Redis-OSS) Cluster anzuwenden, können Sie das AWS CLI oder das Application Auto Scaling API verwenden.

### Anwenden einer Skalierungsrichtlinie mithilfe der AWS CLI

Um eine Skalierungsrichtlinie auf Ihren OSS Cluster ElastiCache mit Valkey oder Redis anzuwenden, verwenden Sie den [put-scaling-policy](#) Befehl mit den folgenden Parametern:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--policy-type` – Legen Sie diesen Wert auf `TargetTrackingScaling` fest.
- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.
- `--target-tracking-scaling-policy-configuration` — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

Im folgenden Beispiel wenden Sie eine Skalierungsrichtlinie `myscalablepolicy` zur Zielverfolgung an, die auf einen ElastiCache mit Valkey oder Redis benannten OSS Cluster mit Auto Scaling benannt `myscalablecluster` ist. ElastiCache Dazu verwenden Sie eine Richtlinienkonfiguration, die in einer Datei mit dem Namen `config.json` gespeichert ist.

Für Linux, macOS oder 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
```

Für 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
```

## Anwenden einer Skalierungsrichtlinie mithilfe von API

Um eine Skalierungsrichtlinie auf Ihren OSS Cluster ElastiCache mit Valkey oder Redis anzuwenden, verwenden Sie den [PutScalingPolicy](#) AWS CLI Befehl mit den folgenden Parametern:

- `—policy-name`– Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.

- -- target-tracking-scaling-policy -configuration — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

Im folgenden Beispiel wenden Sie eine Skalierungsrichtlinie `myscalablepolicy` zur Zielverfolgung an, die auf einen ElastiCache mit Valkey oder Redis benannten OSS Cluster mit Auto Scaling benannt `myscalablecluster` ist. ElastiCache Sie verwenden die Richtlinienkonfiguration `ElastiCachePrimaryEngineCPUUtilization` basierend auf einer vordefinierten Metrik.

```
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"
 }
 }
}
```

## Bearbeiten einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit dem AWS Management Console AWS CLI, oder dem Application Auto Scaling bearbeitenAPI.

## Bearbeiten einer Skalierungsrichtlinie mit dem AWS Management Console

Um eine Auto Scaling Scaling-Richtlinie für einen Cluster ElastiCache mit Valkey oder OSS Redis zu bearbeiten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich die entsprechende Engine aus.
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie unter Scaling policies (Skalierungsrichtlinien) die Schaltfläche links neben der Auto-Scaling-Richtlinie aus, die Sie ändern möchten, und klicken Sie dann auf Modify (Ändern).
6. Nehmen Sie die erforderlichen Änderungen an der Richtlinie vor.
7. Wählen Sie Ändern aus.

## Bearbeiten einer Skalierungsrichtlinie mit dem AWS CLI und API

Sie können das AWS CLI oder das Application Auto Scaling verwendenAPI, um eine Skalierungsrichtlinie auf die gleiche Weise zu bearbeiten, wie Sie eine Skalierungsrichtlinie anwenden:

- Wenn Sie die verwenden AWS CLI, geben Sie den Namen der Richtlinie, die Sie bearbeiten möchten, im `--policy-name` Parameter an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.
- Wenn Sie Application Auto Scaling verwendenAPI, geben Sie den Namen der Richtlinie, die Sie bearbeiten möchten, im `PolicyName` Parameter an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.

Weitere Informationen finden Sie unter [Anwenden einer Skalierungsrichtlinie](#).

## Löschen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit dem AWS Management Console, dem oder dem AWS CLI Application Auto Scaling löschenAPI.

## Löschen einer Skalierungsrichtlinie mit dem AWS Management Console

Um eine Auto Scaling Scaling-Richtlinie für einen ElastiCache (Redis-OSS) Cluster zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder OSSRedis aus.
3. Wählen Sie den Cluster aus, dessen Auto-Scaling-Richtlinie Sie bearbeiten möchten (wählen Sie den Clusternamen aus, nicht die Schaltfläche links davon).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie im Abschnitt Scaling policies (Skalierungsrichtlinien) die Auto-Scaling-Richtlinie und dann Delete (Löschen) aus.

## Löschen einer Skalierungsrichtlinie mit dem AWS CLI

Um eine Skalierungsrichtlinie auf Ihrem ElastiCache WithValkey- oder OSS Redis-Cluster zu löschen, verwenden Sie den [delete-scaling-policy](#) AWS CLI Befehl mit den folgenden Parametern:

- `—policy-name`– Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.

Im folgenden Beispiel löschen Sie eine Skalierungsrichtlinie für die Zielverfolgung mit dem Namen `myscalablepolicy` aus einem Cluster mit dem Namen `myscalablecluster`

Für Linux, macOS oder Unix:

```
aws application-autoscaling delete-scaling-policy \
 --policy-name myscalablepolicy \
 --resource-id replication-group/myscalablecluster \
 --service-namespace elasticache \
 --scalable-dimension elasticache:replication-group:NodeGroups
```

## Für Windows:

```
aws application-autoscaling delete-scaling-policy ^
 --policy-name myscalablepolicy ^
 --resource-id replication-group/myscalablecluster ^
 --service-namespace elasticache ^
 --scalable-dimension elasticache:replication-group:NodeGroups
```

## Löschen einer Skalierungsrichtlinie mit dem API

Um eine Skalierungsrichtlinie auf Ihrem ElastiCache WithValkey- oder OSS Redis-Cluster zu löschen, verwenden Sie den [DeleteScalingPolicy](#) AWS CLI Befehl mit den folgenden Parametern:

- `—policy-name`– Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:NodeGroups`.

Im folgenden Beispiel löschen Sie eine Skalierungsrichtlinie für die Zielverfolgung mit dem Namen `myscalablepolicy` aus einem Cluster mit dem Namen `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"
}
```

## AWS CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden

Dieser Ausschnitt zeigt, wie Sie mithilfe der Ressource::: eine Zielverfolgungsrichtlinie erstellen und sie auf eine [AWSElastiCache:: AWS:ApplicationAutoScaling: ReplicationGroup ScalableTarget](#) -Ressource anwenden. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die ResourceId-Eigenschaft mit dem logischen Namen der `AWS::ElastiCache::ReplicationGroup`-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```
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
```

## Geplante Skalierung

Eine Skalierung anhand eines Zeitplans ermöglicht es Ihnen, Ihre Anwendung entsprechend vorhersagbarer Anforderungsänderungen zu skalieren. Um die geplante Skalierung zu verwenden, erstellen Sie geplante Aktionen, die Valkey oder Redis anweisen ElastiCache, Skalierungsaktivitäten OSS zu bestimmten Zeiten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie einen vorhandenen ElastiCache (Redis-OSS) Cluster an, wann die Skalierungsaktivität stattfinden soll,

sowie die Mindestkapazität und die maximale Kapazität. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Sie können eine geplante Aktion nur für ElastiCache (Redis-OSS) Cluster erstellen, die bereits vorhanden sind. Sie können eine geplante Aktion und einen Cluster nicht gleichzeitig erstellen.

Weitere Informationen zur Terminologie für die Erstellung, Verwaltung und Löschung geplanter Aktionen finden Sie unter [Häufig verwendete Befehle für geplante Aktionen erstellen, verwalten und löschen](#)

So erstellen Sie einen sich wiederholenden Zeitplan:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder OSSRedis aus.
3. Wählen Sie den DB-Cluster aus, zu dem Sie die Richtlinie hinzufügen möchten.
4. Wählen Sie das SymbolVerwalten der Auto Scaling RichtlinieausAktionenDropdown erstellen.
5. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
6. Im Abschnitt Auto-Scaling-Richtlinien wird Skalierungsrichtlinie hinzufügen angezeigt. Klicken Sie auf Geplante Skalierung.
7. Geben Sie unter Policy Name (Richtlinienname) einen Namen für diese Richtlinie ein.
8. Für Skalierbare Dimension wählen Sie Shards aus.
9. Wählen Sie für Ziel-Shards den Wert aus.
10. Für Wiederholung, wählen Sie Wiederkehrende aus.
11. Für Häufigkeit wählen Sie den entsprechenden Wert aus.
12. FürStartdatumundBeginnzeitwählen Sie den Zeitpunkt aus, ab dem die Richtlinie wirksam wird.
13. Wählen Sie Add Policy (Richtlinie hinzufügen) aus.

Erstellen Sie eine einmalige geplante Aktion wie folgt:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder OSSRedis aus.
3. Wählen Sie den DB-Cluster aus, zu dem Sie die Richtlinie hinzufügen möchten.
4. Wählen Sie das SymbolVerwalten der Auto Scaling RichtlinieausAktionenDropdown erstellen.

5. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
6. Im Abschnitt Auto-Scaling-Richtlinien wird Skalierungsrichtlinie hinzufügen angezeigt. Klicken Sie auf Geplante Skalierung.
7. Geben Sie unter Policy Name (Richtliniename) einen Namen für diese Richtlinie ein.
8. Für Skalierbare Dimension wählen Sie Shards aus.
9. Wählen Sie für Ziel-Shards den Wert aus.
10. Wählen Sie für Wiederholung Einmalig aus.
11. Für Startdatum und Beginnzeit wählen Sie den Zeitpunkt aus, ab dem die Richtlinie wirksam wird.
12. Für Enddatum wählen Sie das Datum, bis zu dem die Richtlinie wirksam sein würde.
13. Wählen Sie Add Policy (Richtlinie hinzufügen) aus.

### Löschen einer geplanten Aktion

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder OSSRedis aus.
3. Wählen Sie den DB-Cluster aus, zu dem Sie die Richtlinie hinzufügen möchten.
4. Wählen Sie das Symbol Verwalten der Auto Scaling Richtlinie aus Aktionen Dropdown erstellen.
5. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
6. Wählen Sie im Abschnitt Auto-Scaling-Richtlinien die Auto-Scaling-Richtlinie und dann Löschen im Aktionen-Dialog aus.

### Verwalten einer geplanten Skalierung mit AWS CLI

Verwenden Sie die folgende automatische Anwendungsskalierung: APIs

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Verwenden Sie AWS CloudFormation für die Erstellung einer geplanten Aktion

Dieser Ausschnitt zeigt, wie Sie mithilfe der Ressource::: eine Zielverfolgungsrichtlinie erstellen und sie auf eine [AWSElastiCache:: AWS:ApplicationAutoScaling: ReplicationGroup](#)

[ScalableTarget](#) -Ressource anwenden. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die ResourceId-Eigenschaft mit dem logischen Namen der `AWS::ElastiCache::ReplicationGroup`-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```
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 * * ? *)'
```

## Verwenden von Auto Scaling mit Replikaten

Eine ElastiCache Replikationsgruppe kann einen oder mehrere Caches so einrichten, dass sie als einzelner logischer Knoten funktionieren.

Im Folgenden finden Sie Einzelheiten zur Zielverfolgung und zu geplanten Richtlinien sowie deren Anwendung mithilfe von AWS Management Console AWS CLI und APIs.

### Skalierungsrichtlinien für die Ziel-Nachverfolgung

Mit Skalierungsrichtlinien für die Zielverfolgung wählen Sie eine Metrik aus und legen einen Zielwert fest. ElastiCache mit Valkey oder Redis werden die CloudWatch Alarme OSS AutoScaling erstellt und verwaltet, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung auf der Grundlage der Metrik und des Zielwerts. Die Skalierungsrichtlinie fügt -Repliken hinzu oder entfernt sie, wenn erforderlich, um die Metrik auf dem angegebenen Zielwert oder in der Nähe davon zu halten. Abgesehen davon, dass eine Skalierungsrichtlinie für die Ziel-Nachverfolgung die Metrik

nahe an dem Zielwert hält, passt sie sich auch an die Schwankungen in der Metrik aufgrund eines schwankenden Lastmusters an und verringert schnelle Schwankungen der Kapazität der Flotte.

## Kriterien für die Auto Scaling von Replikaten

Ihre Auto Scaling Richtlinie definiert die folgende vordefinierte Metrik für Ihren Cluster:

`ElastiCacheReplicaEngineCPUUtilization`: Der Schwellenwert für die AVG CPU Engine-Auslastung, aggregiert für alle Replikate, der ElastiCache verwendet wird, um einen auto-scaling Skalierungsvorgang auszulösen. Sie können das Auslastungsziel zwischen 35 Prozent und 70 Prozent festlegen.

Wenn der Service feststellt, dass Ihre `ElastiCacheReplicaEngineCPUUtilization` Metrik gleich oder größer als die Zieleinstellung ist, erhöht er automatisch die Anzahl der Replikate auf Ihren Shards. ElastiCache mit Valkey oder Redis werden Ihre Cluster-Replikate um eine Anzahl OSS skaliert, die der größeren von zwei Zahlen entspricht: prozentuale Abweichung von Target und einem Replikate. Bei der Skalierung ElastiCache mit Valkey oder Redis erfolgt OSS keine auto Skalierung, es sei denn, der Gesamtwert liegt unter 75 Prozent Ihres definierten Ziels.

Wenn Sie für ein Scale-Out-Beispiel jeweils 5 Shards und 1 Replikate haben:

Wenn Ihr Target die Sicherheitslücke um 30 Prozent überschreitet, wird ElastiCache mit Valkey oder Redis ein Replikate (max. (0,3, Standard 1)) für alle Shards OSS skaliert. Das Ergebnis sind 5 Shards mit jeweils 2 Replikaten.

Wenn Sie beispielsweise einen Zielwert von 60 Prozent ausgewählt haben, ElastiCache mit Valkey oder Redis OSS wird die auto Skalierung erst dann automatisch durchgeführt, wenn die Metrik 45 Prozent oder weniger beträgt (25 Prozent unter dem Ziel von 60 Prozent).

## Überlegungen zum Auto Scaling

Beachten Sie folgende Überlegungen:

- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung geht davon aus, dass sie eine horizontale Skalierung nach oben vornehmen soll, wenn die angegebene Metrik über dem Zielwert liegt. Sie können keine Skalierungsrichtlinie für die Zielverfolgung verwenden, um eine Skalierung vorzunehmen, wenn die angegebene Metrik unter dem Zielwert liegt. ElastiCache mit Valkey oder Redis werden Replikate auf alle Shards im Cluster um das Maximum (% Abweichung abgerundet von Target, Standard 1) vorhandener Replikate OSS skaliert.

- Eine Skalierungsrichtlinie für die Ziel-Nachverfolgung nimmt keine Skalierung vor, wenn die angegebene Metrik unzureichende Daten aufweist. Es wird keine horizontale Skalierung nach unten vorgenommen, da unzureichende Daten nicht als geringe Auslastung interpretiert werden.
- Möglicherweise werden Lücken zwischen den Datenpunkten für den Zielwert und die aktuelle Metrik angezeigt. Das liegt daran, dass OSS Auto Scaling ElastiCache bei Valkey oder Redis immer konservativ agiert, indem es auf- oder abrundet, wenn es bestimmt, wie viel Kapazität hinzugefügt oder entfernt werden soll. Dadurch wird verhindert, dass zu wenig Kapazität hinzugefügt oder zu viel Kapazität entfernt wird.
- Um die Verfügbarkeit der Anwendung sicherzustellen, wird der Service schnellstmöglich proportional zur Metrik hochskaliert, jedoch etwas langsamer herunterskaliert.
- Sie können mehrere Skalierungsrichtlinien für die Zielverfolgung für einen OSS Cluster ElastiCache mit Valkey oder Redis verwenden, vorausgesetzt, dass jede von ihnen eine andere Metrik verwendet. Die Absicht von Auto Scaling besteht darin, der Verfügbarkeit immer Priorität einzuräumen. Daher unterscheidet sich das Verhalten je nachdem, ob die Zielverfolgungsrichtlinien für die horizontale Skalierung oder für die Skalierung bereit sind. Sofern Richtlinien für die Ziel-Nachverfolgung für die Skalierung nach oben bereit sind, findet eine Skalierung des Service nach oben statt. Eine Skalierung nach unten wird jedoch nur vorgenommen, wenn alle Richtlinien für die Ziel-Nachverfolgung (mit aktivierter Skalierung nach unten) zur Skalierung nach unten bereit sind.
- Bearbeiten oder löschen Sie nicht die CloudWatch Alarme, die ElastiCache mit Valkey oder Redis OSS Auto Scaling für eine Skalierungsrichtlinie zur Zielverfolgung verwaltet werden. Auto Scaling löscht die Alarme automatisch, wenn Sie die Skalierungsrichtlinie oder den Cluster löschen.
- ElastiCache mit Valkey oder Redis verhindert OSS Auto Scaling nicht, dass Sie Replikate auf mehreren Shards manuell ändern. Diese manuellen Anpassungen wirken sich nicht auf bestehende CloudWatch Alarme aus, die mit der Skalierungsrichtlinie verknüpft sind, können sich jedoch auf Metriken auswirken, die diese Alarme auslösen können. CloudWatch
- Diese von Auto Scaling verwalteten CloudWatch Alarme werden über die AVG Metrik für alle Shards im Cluster definiert. So kann Hot-Shards zu einem beliebigen Szenario führen:
  - Skalierung, wenn sie aufgrund der Belastung einiger Hot-Shards, die einen Alarm auslösen, nicht erforderlich ist CloudWatch
  - wird nicht skaliert, wenn dies erforderlich ist, da alle Shards aggregiert werden, AVG was dazu führt, dass der Alarm nicht überschritten wird.
- ElastiCache Bei Valkey oder Redis gelten weiterhin die OSS Standardgrenzwerte für Knoten pro Cluster. Wenn Sie sich also für Auto Scaling entscheiden und erwarten, dass maximale Knoten mehr als die Standardgrenze sind, fordern Sie eine Limiterhöhung bei [AWS -Service-Limits](#) an und wählen Sie den Limit-Typ Knoten pro Cluster pro Instance-Typ.

- Stellen Sie sicher, dass in Ihrem System genügend ENIs (Elastic Network Interfaces) zur Verfügung stehen VPC, die beim Scale-Out benötigt werden. Weitere Informationen finden Sie unter [Elastic-Network-Schnittstellen](#).
- Wenn nicht genügend Kapazität verfügbar ist EC2, würde OSS Auto Scaling ElastiCache mit Valkey oder Redis erst skalieren, wenn die Kapazität verfügbar ist oder wenn Sie den Cluster manuell auf die Instance-Typen mit ausreichender Kapazität ändern.
- ElastiCache mit Valkey oder Redis unterstützt OSS Auto Scaling keine Skalierung von Replikaten mit einem Cluster mit `ReservedMemoryPercent` weniger als 25 Prozent. Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

### Hinzufügen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie hinzufügen, indem Sie die verwenden AWS Management Console.

### Hinzufügen einer Skalierungsrichtlinie mithilfe des AWS Management Console

Um ElastiCache mit Valkey oder Redis eine Auto Scaling-Richtlinie hinzuzufügen OSS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder OSS Redis aus.
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie Add dynamic scaling (Dynamische Skalierung hinzufügen) aus.
6. Wählen Sie unter Scaling policies (Skalierungsrichtlinien) Add dynamic scaling (Dynamische Skalierung hinzufügen) aus.
7. Geben Sie unter Policy Name (Richtliniennamen) einen Namen für diese Richtlinie ein.
8. Für Skalierbare Dimension auf und danach Replicas-Dialogfeld aus erstellen.
9. Geben Sie als Zielwert den durchschnittlichen Prozentsatz der CPU Auslastung ein, den Sie für Replikate beibehalten möchten. ElastiCache Dieser Wert muss größer 35 und kleiner 70 sein. - Replikaten werden hinzugefügt oder entfernt, um die Metrik in der Nähe des angegebenen Wertes zu halten.
10. (Optional) Die Ruhephase für Scale-In- oder Scale-Out-Ruhephase werden von der Konsole nicht unterstützt. Verwenden Sie den AWS CLI , um die Abkühlwerte zu ändern.

11. Geben Sie für Mindestkapazität die Mindestanzahl von Replikaten ein, für deren Wartung die ElastiCache with Valkey- oder Redis OSS Auto Scaling Scaling-Richtlinie erforderlich ist.
12. Geben Sie für Maximale Kapazität die maximale Anzahl von Replikaten ein, für deren Wartung die ElastiCache with Valkey- oder Redis OSS Auto Scaling Scaling-Richtlinie erforderlich ist. Der Wert muss größer als 5 sein.
13. Wählen Sie Create (Erstellen) aus.

## Registrieren eines skalierbaren Ziels

Sie können eine Skalierungsrichtlinie anwenden, die entweder auf einer vordefinierten oder einer benutzerdefinierten Metrik basiert. Dazu können Sie das AWS CLI oder das Application Auto Scaling verwenden. Der erste Schritt besteht darin, Ihre OSS Replikationsgruppe ElastiCache bei Valkey oder Redis bei Auto Scaling zu registrieren.

Bevor Sie ElastiCache Auto Scaling mit einem Cluster verwenden können, müssen Sie Ihren Cluster ElastiCache bei Valkey oder Redis OSS Auto Scaling registrieren. Sie tun dies, um die Skalierungsdimension und die Grenzwerte zu definieren, die auf diesen Cluster angewendet werden sollen. ElastiCache mit Valkey oder Redis skaliert OSS Auto Scaling den Cluster dynamisch entlang der `elasticache:replication-group:Replicas` skalierbaren Dimension, die die Anzahl der Cluster-Replikat pro Shard darstellt.

## Unter Verwendung der CLI

Verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern, um Ihren ElastiCache Cluster zu registrieren:

- `--service-namespace` — Legen Sie diesen Wert auf `elasticache` fest.
- `--resource-id` — Die Ressourcen-ID für den Cluster. ElastiCache Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters. `replication-group/myscalablecluster`
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.
- `--min-capacity` — Die Mindestanzahl von Replikaten, die ElastiCache mit Valkey oder Redis Auto Scaling verwaltet werden sollen. OSS Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikat in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

- `--max-capacity` — Die maximale Anzahl von Replikaten, die ElastiCache mit Valkey oder Redis Auto Scaling verwaltet werden können. OSS Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

## Example

Im folgenden Beispiel registrieren Sie bei Valkey oder Redis einen Cluster ElastiCache mit dem Namen `myscalablecluster`. Die Registrierung zeigt an, dass der Cluster dynamisch skaliert werden soll, um zwischen einem und 5 Replikaten zu haben.

Für Linux, macOS oder 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 \

```

Für 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 ^\

```

## Mit dem API

Verwenden Sie den [register-scalable-target](#) Befehl mit den folgenden Parametern, um Ihren ElastiCache Cluster zu registrieren:

- `ServiceNamespace` — Setzen Sie diesen Wert auf `elasticache`.
- `ResourceID` — Die Ressourcen-ID für den ElastiCache Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise `replication-group/myscalablecluster` der Name des Clusters.
- `ScalableDimension` — Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.

- **MinCapacity** — Die Mindestanzahl von Replikaten, die ElastiCache mit Valkey oder Redis OSS Auto Scaling verwaltet werden müssen. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).
- **MaxCapacity** — Die maximale Anzahl von Replikaten, die ElastiCache mit Valkey oder Redis OSS Auto Scaling verwaltet werden können. Informationen zur Beziehung zwischen `--min-capacity`, `--max-capacity` und der Anzahl der Replikate in Ihrem Cluster finden Sie unter [Minimale und maximale Kapazität](#).

## Example

Im folgenden Beispiel registrieren Sie einen Cluster `myscalablecluster` mit dem Namen `Application Auto ScalingAPI`. Diese Registrierung zeigt an, dass der Cluster dynamisch skaliert werden soll, um zwischen 1 und 5 Replikate zu haben.

```
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
}
```

## Definieren einer Skalierungsrichtlinie

Eine Konfiguration einer Skalierungsrichtlinie zur Zielverfolgung wird durch einen JSON Block dargestellt, in dem die Metriken und Zielwerte definiert sind. Sie können eine Konfiguration der Skalierungsrichtlinie als JSON Block in einer Textdatei speichern. Sie verwenden diese Textdatei, wenn Sie die AWS CLI oder die Application Auto Scaling API aufrufen. Weitere Informationen zur Syntax der Richtlinienkonfiguration finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der APIReferenz zu Application Auto Scaling.

Die folgenden Optionen stehen zur Verfügung, um eine Konfiguration der Skalierungsrichtlinien für die Zielverfolgung zu definieren:

#### Themen

- [Verwenden einer vordefinierten Metrik](#)
- [Bearbeiten einer Skalierungsrichtlinie](#)
- [Löschen einer Skalierungsrichtlinie](#)
- [AWS CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden](#)
- [Geplante Skalierung](#)

#### Verwenden einer vordefinierten Metrik

Eine Konfiguration einer Skalierungsrichtlinie zur Zielverfolgung wird durch einen JSON Block dargestellt, in dem die Metriken und Zielwerte definiert sind. Sie können eine Konfiguration der Skalierungsrichtlinie als JSON Block in einer Textdatei speichern. Sie verwenden diese Textdatei, wenn Sie die AWS CLI oder die Application Auto Scaling API aufrufen. Weitere Informationen zur Syntax der Richtlinienkonfiguration finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der APIReferenz zu Application Auto Scaling.

Die folgenden Optionen stehen zur Verfügung, um eine Konfiguration der Skalierungsrichtlinien für die Zielverfolgung zu definieren:

#### Themen

- [Verwenden einer vordefinierten Metrik](#)
- [Verwenden einer benutzerdefinierten Metrik](#)
- [Verwenden von Ruhephasen](#)
- [Deaktivieren der Scale-In-Aktivität](#)
- [Anwendung einer Skalierungsrichtlinie auf einen Cluster ElastiCache mit Valkey oder Redis OSS](#)

#### Verwenden einer vordefinierten Metrik

Mithilfe vordefinierter Metriken können Sie schnell eine Skalierungsrichtlinie für die Zielverfolgung für einen OSS Cluster ElastiCache mit Valkey oder Redis definieren, der mit Zielverfolgung in ElastiCache Valkey oder Redis Auto Scaling funktioniert. OSS ElastiCache unterstützt derzeit die folgende vordefinierte Metrik in ElastiCache Replicas Auto Scaling:

**ElastiCacheReplicaEngineCPUUtilization**— Der Durchschnittswert der `ngineCPUUtilization` E-Metrik CloudWatch für alle Replikate im Cluster. Den aggregierten Metrikwert finden Sie CloudWatch unter „Erforderlich“ `ReplicationGroupId` und „ElastiCache `ReplicationGroupId`, `RoleRollenreplikate`“.

Um eine vordefinierte Metrik in Ihrer Skalierungsrichtlinie zu verwenden, erstellen Sie eine Zielverfolgungskonfiguration für Ihre Skalierungsrichtlinie. Diese Konfiguration muss eine `PredefinedMetricSpecification` für die vordefinierte Metrik und einen `TargetValue` für den Zielwert dieser Metrik enthalten.

### Verwenden einer benutzerdefinierten Metrik

Durch die Verwendung von benutzerdefinierten Metriken können Sie eine Skalierungsrichtlinie für die Zielverfolgung definieren, die Ihren individuellen Anforderungen entspricht. Sie können eine benutzerdefinierte Metrik definieren, die auf einer beliebigen Metrik ElastiCache mit Valkey oder Redis basiert und sich OSS proportional zur Skalierung ändert. Nicht alle ElastiCache Metriken eignen sich für die Zielverfolgung. Die Metrik muss eine gültige Auslastungsmetrik sein und beschreiben, wie ausgelastet eine Instance ist. Der Wert der Metrik muss sich proportional zur Anzahl der `-Replicas` im `-DB-Cluster` erhöhen oder verringern. Diese proportionale Erhöhung oder Verminderung ist notwendig, um die metrischen Daten zur proportionalen Skalierung oder in der Anzahl der `-Repliken` zu verwenden.

### Example

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt eine benutzerdefinierte Metrik einen Cluster auf der Grundlage einer durchschnittlichen CPU Auslastung von 50 Prozent aller Replikate in einem Cluster mit dem Namen `an.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"
 }
}
```

```
}
```

## Verwenden von Ruhephasen

Sie können einen Wert in Sekunden für `ScaleOutCooldown` angeben, um eine Ruhephase für die Aufskalierung Ihres Clusters hinzuzufügen. Ähnlich können Sie einen Wert in Sekunden für `ScaleInCooldown` angeben, um eine Ruhephase für die Abskalierung Ihres Clusters hinzuzufügen. Weitere Informationen zu `ScaleInCooldown` und `ScaleOutCooldown` finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der APIReferenz zu Application Auto Scaling. Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration wird die `ElastiCacheReplicaEngineCPUUtilization` vordefinierte Metrik verwendet, um einen Cluster auf der Grundlage einer durchschnittlichen CPU Auslastung von 40 Prozent aller Replikate in diesem Cluster anzupassen. Die Konfiguration sieht eine Scale-In-Ruhephase von 10 Minuten und eine Scale-Out-Ruhephase von 5 Minuten vor.

```
{"TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
 },
 "ScaleInCooldown": 600,
 "ScaleOutCooldown": 300
}
```

## Deaktivieren der Scale-In-Aktivität

Sie können verhindern, dass die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung in Ihrem OSS Cluster ElastiCache mit Valkey oder Redis skaliert, indem Sie die Scale-In-Aktivität deaktivieren. Das Deaktivieren der Scale-In-Aktivität verhindert das Löschen von -Repliken durch die Skalierungsrichtlinie, und erlaubt der Skalierungsrichtlinie dennoch, Repliken nach Bedarf zu erstellen.

Sie können einen booleschen Wert für `DisableScaleIn` angeben, um die Scale-In-Aktivität für Ihren Cluster zu aktivieren oder zu deaktivieren. Weitere Informationen zu `DisableScaleIn` finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der APIReferenz zu Application Auto Scaling.

## Example

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt die `ElastiCacheReplicaEngineCPUUtilization` vordefinierte Metrik einen Cluster auf der Grundlage einer durchschnittlichen CPU Auslastung von 40 Prozent

aller Replikate in diesem Cluster an. Die Konfiguration deaktiviert die Scale-In-Aktivität für die Skalierungsrichtlinie.

```
{"TargetValue": 40.0,
 "PredefinedMetricSpecification":
 {"PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
 },
 "DisableScaleIn": true
}
```

Anwendung einer Skalierungsrichtlinie auf einen Cluster ElastiCache mit Valkey oder Redis OSS

Nachdem Sie Ihren Cluster ElastiCache bei Valkey oder Redis OSS Auto Scaling registriert und eine Skalierungsrichtlinie definiert haben, wenden Sie die Skalierungsrichtlinie auf den registrierten Cluster an. Um eine Skalierungsrichtlinie auf einen OSS Cluster ElastiCache mit Valkey oder Redis anzuwenden, können Sie das AWS CLI oder das Application Auto Scaling verwenden. API

Mit dem AWS CLI

Verwenden Sie den [put-scaling-policy](#) Befehl ElastiCache mit den folgenden Parametern, um eine Skalierungsrichtlinie auf Ihren OSS Cluster mit Valkey oder Redis anzuwenden:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--policy-type` – Setzen Sie diesen Wert auf `TargetTrackingScaling`.
- `--resource-id` — Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters.  
`replication-group/myscalablecluster`
- `—service-namespace` — Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.
- `--target-tracking-scaling-policy-configuration` — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

## Example

Im folgenden Beispiel wenden Sie eine Zielverfolgungs-Skalierungsrichtlinie an, die auf einen Cluster benannt ist `myscalablepolicy`, der `myscalablecluster` ElastiCache mit Valkey oder OSS Redis Auto Scaling benannt ist. Dazu verwenden Sie eine Richtlinienkonfiguration, die in einer Datei mit dem Namen `config.json` gespeichert ist.

## Für Linux, macOS oder 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
}
```

## Für 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
```

## Verwenden Sie den API

Um eine Skalierungsrichtlinie auf Ihren OSS Cluster ElastiCache mit Valkey oder Redis mit Application Auto Scaling anzuwenden, verwenden Sie den [PutScalingPolicy](#) Application Auto Scaling API Scaling-Vorgang mit den folgenden Parametern:

- **PolicyName** — Der Name der Skalierungsrichtlinie.
- **PolicyType** — Setzen Sie diesen Wert auf `TargetTrackingScaling`.
- **ResourceID** — Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des ElastiCache (Redis-OSS) Clusters. `replication-group/myscalablecluster`

- `ServiceNamespace` — Setzen Sie diesen Wert auf `Elasticache`.
- `ScalableDimension` — Setze diesen Wert auf `elasticache:replication-group:Replicas`.
- `TargetTrackingScalingPolicyConfiguration` — Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den Cluster verwendet werden soll.

## Example

Im folgenden Beispiel wenden Sie eine Skalierungsrichtlinie für die Zielverfolgung mit dem Namen auf einen Cluster `scalablepolicy` an, der `myscalablecluster` ElastiCache mit Valkey oder OSS Redis Auto Scaling benannt ist. Sie verwenden die Richtlinienkonfiguration `ElastiCacheReplicaEngineCPUUtilization` basierend auf einer vordefinierten Metrik.

```
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"
 }
 }
}
```

## Bearbeiten einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit dem AWS Management Console AWS CLI, oder dem Application Auto Scaling bearbeitenAPI.

## Bearbeiten einer Skalierungsrichtlinie mit dem AWS Management Console

Sie können Richtlinien mit dem Typ Vordefinierte Metriken nur bearbeiten, indem Sie die AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS
3. Wählen Sie den Cluster aus, dem Sie eine Richtlinie hinzufügen möchten (wählen Sie den Cluster-Namen und nicht die Schaltfläche links davon aus).
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie unter Scaling policies (Skalierungsrichtlinien) die Schaltfläche links neben der Auto-Scaling-Richtlinie aus, die Sie ändern möchten, und klicken Sie dann auf Modify (Ändern).
6. Nehmen Sie die erforderlichen Änderungen an der Richtlinie vor.
7. Wählen Sie Ändern aus.
8. Nehmen Sie die Änderungen an der Richtlinie vor.
9. Wählen Sie Ändern aus.

## Bearbeiten einer Skalierungsrichtlinie mithilfe von AWS CLI oder der Application Auto Scaling API

Sie können das AWS CLI oder das Application Auto Scaling verwendenAPI, um eine Skalierungsrichtlinie auf die gleiche Weise zu bearbeiten, wie Sie eine Skalierungsrichtlinie anwenden:

- Wenn Sie Application Auto Scaling verwendenAPI, geben Sie den Namen der Richtlinie, die Sie bearbeiten möchten, im PolicyName Parameter an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.

Weitere Informationen finden Sie unter [Anwendung einer Skalierungsrichtlinie auf einen Cluster ElastiCache mit Valkey oder Redis OSS](#).

## Löschen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit dem AWS Management Console, dem AWS CLI oder dem Application Auto Scaling löschen API

## Löschen einer Skalierungsrichtlinie mit dem AWS Management Console

Sie können Richtlinien mit dem Typ Vordefinierte Metriken nur bearbeiten, indem Sie die AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS
3. Wählen Sie den Cluster aus, dessen Auto-Scaling-Richtlinie Sie löschen möchten.
4. Wählen Sie die Registerkarte Auto-Scaling-Richtlinien aus.
5. Wählen Sie im Abschnitt Scaling policies (Skalierungsrichtlinien) die Auto-Scaling-Richtlinie und dann Delete (Löschen) aus.

## Löschen einer Skalierungsrichtlinie mithilfe von AWS CLI oder der Application Auto Scaling API

Sie können das AWS CLI oder das Application Auto Scaling verwendenAPI, um eine Skalierungsrichtlinie aus einem ElastiCache Cluster zu löschen.

### CLI

Um eine Skalierungsrichtlinie aus Ihrem ElastiCache WithValkey- oder OSS Redis-Cluster zu löschen, verwenden Sie den [delete-scaling-policy](#) Befehl mit den folgenden Parametern:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des Clusters.  
`replication-group/myscalablecluster`
- `—service-namespace` — Legen Sie diesen Wert auf `elasticache` fest.
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.

### Example

Im folgenden Beispiel löschen Sie eine Skalierungsrichtlinie für die Zielverfolgung mit dem Namen eines Clusters mit `myscalablepolicy` dem Namen `ELC.myscalablecluster`

Für Linux, macOS oder Unix:

```
aws application-autoscaling delete-scaling-policy \
 --policy-name myscalablepolicy \
 --resource-id replication-group/myscalablecluster \
 --service-namespace elasticache \
 --scalable-dimension elasticache:replication-group:Replicas \

```

Für 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

Verwenden Sie den [DeleteScalingPolicy](#) Application Auto API Scaling-Vorgang ElastiCache mit den folgenden Parametern, um eine Skalierungsrichtlinie aus Ihrem OSS Cluster mit Valkey oder Redis zu löschen:

- **PolicyName** — Der Name der Skalierungsrichtlinie.
- **ResourceID** — Die Ressourcen-ID für den Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise `replication-group/myscalablecluster` der Name des Clusters.
- **ServiceNamespace** — Setzen Sie diesen Wert auf `Elasticache`.
- **ScalableDimension** — Setze diesen Wert auf `elasticache:replication-group:Replicas`.

Im folgenden Beispiel löschen Sie eine Skalierungsrichtlinie für die Zielverfolgung, die `myscalablepolicy` aus einem Cluster benannt ist, der `myscalablecluster` mit Application Auto Scaling benannt ist. 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"
}
```

## AWS CloudFormation Für Auto Scaling Scaling-Richtlinien verwenden

Dieser Ausschnitt zeigt, wie Sie mithilfe der Ressource [AWS::: eine geplante Aktion erstellen und sie auf eineElastiCache:: AWS:ApplicationAutoScaling: ReplicationGroup ScalableTarget](#) -Ressource anwenden. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die ResourceId-Eigenschaft mit dem logischen Namen der `AWS::ElastiCache::ReplicationGroup`-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```
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

## Geplante Skalierung

Eine Skalierung anhand eines Zeitplans ermöglicht es Ihnen, Ihre Anwendung entsprechend vorhersagbarer Anforderungsänderungen zu skalieren. Um die geplante Skalierung zu verwenden, erstellen Sie geplante Aktionen, die Valkey oder Redis anweisen ElastiCache, Skalierungsaktivitäten OSS zu bestimmten Zeiten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie einen vorhandenen OSS Cluster ElastiCache mit Valkey oder Redis an, wann die Skalierungsaktivität stattfinden soll, sowie die Mindestkapazität und die maximale Kapazität. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Sie können eine geplante Aktion nur für ElastiCache bereits vorhandene Valkey- oder OSS Redis-Cluster erstellen. Sie können eine geplante Aktion und einen Cluster nicht gleichzeitig erstellen.

Weitere Informationen zur Terminologie für die Erstellung, Verwaltung und Löschung geplanter Aktionen finden Sie unter [Häufig verwendete Befehle für geplante Aktionen erstellen, verwalten und löschen](#)

Erstellen Sie eine einmalige geplante Aktion wie folgt

Ähnlich wie Shard Dimension. Siehe [Geplante Skalierung](#).

Löschen einer geplanten Aktion

Ähnlich wie Shard Dimension. Siehe [Geplante Skalierung](#).

Verwalten einer geplanten Skalierung mit AWS CLI

Verwenden Sie die folgende automatische Anwendungsskalierung: APIs

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Wird verwendet AWS CloudFormation, um Auto Scaling Scaling-Richtlinien zu erstellen

Dieser Ausschnitt zeigt, wie Sie mithilfe der Ressource [AWS::: eine geplante Aktion erstellen und sie auf eineElastiCache:: AWS:ApplicationAutoScaling: ReplicationGroup ScalableTarget](#) -Ressource

anwenden. Sie verwendet die intrinsischen Funktionen [Fn::Join](#) und [Ref](#), um die ResourceId-Eigenschaft mit dem logischen Namen der `AWS::ElastiCache::ReplicationGroup`-Ressource zu erstellen, die in derselben Vorlage angegeben ist.

```
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 * * ? *)'
```

## Ändern des Clustermodus

Valkey und Redis OSS sind verteilte In-Memory-Datenbanken, die Sharding und Replikation unterstützen. ElastiCache Valkey- und OSS Redis-Cluster sind die verteilte Implementierung, mit der Daten auf mehrere Knoten partitioniert werden können. Ein ElastiCache (Redis-OSS) Cluster hat zwei Betriebsmodi: den Clustermodus aktiviert (CME) und den Clustermodus deaktiviert (). CMD In CME arbeitet eine Valkey- und OSS Redis-Engine als verteilte Datenbank mit mehreren Shards und Knoten, während Valkey und Redis OSS in CMD als ein einziger Knoten arbeiten.

Vor der Migration von CMD zu müssen die folgenden Bedingungen CME erfüllt sein:

### Important

Die Konfiguration des Clustermodus kann nur vom deaktivierten in den aktivierten Clustermodus geändert werden. Das Zurücksetzen dieser Konfiguration ist nicht möglich.

- Der Cluster darf nur Schlüssel in der Datenbank 0 haben.
- Anwendungen müssen einen Valkey- oder OSS Redis-Client verwenden, der das Cluster-Protokoll verwenden kann, und einen Konfigurationsendpunkt verwenden.
- Das automatische Failover muss auf dem Cluster mit mindestens einem Replikat aktiviert sein.
- Die für die Migration erforderliche Engine-Version ist mindestens Valkey 7.2 und höher oder Redis OSS 7.0 und höher.

Um von CMD zu migrierenCME, muss die Konfiguration des Clustermodus von deaktiviertem Clustermodus auf aktivierten Clustermodus geändert werden. Dies ist ein zweistufiges Verfahren, das die Clusterverfügbarkeit während des Migrationsprozesses sicherstellt.

### Note

Sie müssen eine Parametergruppe mit clusterfähiger Konfiguration angeben, d. h., der clusterfähige Parameter ist auf `yes` festgelegt. Wenn Sie eine Standardparametergruppe verwenden, wählt ElastiCache (RedisOSS) automatisch die entsprechende Standardparametergruppe mit einer clusterfähigen Konfiguration aus. Der Cluster-aktivierte Parameterwert ist für einen Cluster auf `gesetzt`. `no` CMD Wenn der Cluster in den kompatiblen Modus wechselt, wird der clusterfähige Parameterwert im Rahmen der Änderungsaktion auf `yes` aktualisiert.

Weitere Informationen finden Sie unter [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#)

1. Vorbereiten — Erstellen Sie einen CME Testcluster und stellen Sie sicher, dass Ihr Stack bereit ist, damit zu arbeiten. ElastiCache (RedisOSS) hat keine Möglichkeit, Ihre Bereitschaft zu überprüfen. Weitere Informationen finden Sie unter [Einen Cluster für Valkey oder Redis erstellen OSS](#).
2. Ändern Sie die bestehende CMD Clusterkonfiguration so, dass sie mit dem Clustermodus kompatibel ist — In diesem Modus wird ein einziger Shard bereitgestellt, und ElastiCache (RedisOSS) funktioniert als einzelner Knoten, aber auch als einzelner Shard-Cluster. Kompatibler Modus bedeutet, dass die Client-Anwendung eines der beiden Protokolle verwenden kann, um mit dem Cluster zu kommunizieren. In diesem Modus müssen Anwendungen neu konfiguriert werden, damit sie das Valkey- oder OSS Redis-Cluster-Protokoll und den Konfigurationsendpunkt verwenden können. Gehen Sie wie folgt vor, um den Valkey- oder OSS Redis-Clustermodus auf einen kompatiblen Clustermodus umzustellen:

**Note**

Im kompatiblen Modus sind andere Änderungsvorgänge, wie Änderung der Skalierung und der Engine-Version, für den Cluster nicht zulässig. Darüber hinaus können Parameter (ausgenommen `cacheParameterGroupName`) nicht geändert werden, wenn der Clustermodus-Parameter in der Anfrage definiert wird. [ModifyReplicationGroup](#)

- a. Verwenden Sie den AWS Management Console, sehen Sie sich den Clustermodus an [Ändern einer Replikationsgruppe](#) und setzen Sie ihn auf `Compatible`
- b. Verwenden Sie den API, sehen Sie sich den `ClusterMode` Parameter an [ModifyReplicationGroup](#) und aktualisieren Sie ihn auf `compatible`.
- c. Verwenden Sie den AWS CLI, sehen Sie sich den `cluster-mode` Parameter an [modify-replication-group](#) und aktualisieren Sie ihn auf `compatible`.

Nach dem Ändern des Valkey- oder OSS Redis-Clustermodus in den Clustermodus kompatibel, [DescribeReplicationGroups](#) API wird der ElastiCache (Redis-OSS) Clusterkonfigurationsendpunkt zurückgegeben. Der Cluster-Konfigurationsendpunkt ist ein einzelner Endpunkt, der von Anwendungen verwendet werden kann, um eine Verbindung mit dem Cluster herzustellen. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

3. Clusterkonfiguration in den aktivierten Clustermodus ändern – Sobald der Clustermodus auf den kompatiblen Clustermodus eingestellt ist, besteht der zweite Schritt darin, die Clusterkonfiguration so zu ändern, dass der Clustermodus aktiviert ist. In diesem Modus wird ein einzelner Shard ausgeführt und Kunden können jetzt ihre Cluster skalieren oder andere Clusterkonfigurationen ändern.

Gehen Sie wie folgt vor, um den Clustermodus in aktiviert zu ändern:

Bevor Sie beginnen, stellen Sie sicher, dass Ihre Valkey- oder OSS Redis-Clients auf das Cluster-Protokoll umgestellt wurden und dass der Konfigurationsendpunkt des Clusters nicht verwendet wird.

- a. Sehen Sie sich mit dem den AWS Management Console Clustermodus an [Ändern einer Replikationsgruppe](#) und setzen Sie ihn auf `Aktiviert`.
- b. Verwenden Sie den API, sehen Sie sich den `ClusterMode` Parameter an [ModifyReplicationGroup](#) und aktualisieren Sie ihn auf `enabled`.

- c. Verwenden Sie den AWS CLI, sehen Sie sich den `cluster-mode` Parameter an [modify-replication-group](#) und aktualisieren Sie ihn auf `enabled`.

Nachdem der Clustermodus auf aktiviert geändert wurde, werden die Endpoints gemäß der Valkey- oder OSS Redis-Clusterspezifikation konfiguriert. Das gibt [DescribeReplicationGroups](#) API den Clustermodus-Parameter als `enabled` und die Cluster-Endpunkte zurück, die jetzt für Anwendungen zur Verbindung mit dem Cluster verfügbar sind.

Beachten Sie, dass sich die Cluster-Endpunkte ändern, sobald der Clustermodus auf aktiviert geändert wird. Achten Sie darauf, Ihre Anwendungen mit den neuen Endpunkten zu aktualisieren.

Sie können sich auch dafür entscheiden, aus dem kompatiblen Clustermodus wieder zum Clustermodus deaktiviert (CMD) zurückzukehren und die ursprünglichen Konfigurationen beizubehalten.

Ändern der Clusterkonfiguration vom kompatiblen in den deaktivierten Clustermodus

1. Wenn Sie den Clustermodus verwenden AWS Management Console, sehen Sie sich den Clustermodus an [Ändern einer Replikationsgruppe](#) und setzen Sie ihn auf Deaktiviert
2. Verwenden Sie den API, sehen Sie sich den `ClusterMode` Parameter an [ModifyReplicationGroup](#) und aktualisieren Sie ihn auf `disabled`.
3. Verwenden Sie den AWS CLI, sehen Sie sich den `cluster-mode` Parameter an [modify-replication-group](#) und aktualisieren Sie ihn auf `disabled`.

Nachdem Sie den Clustermodus auf deaktiviert geändert haben, gibt der [DescribeReplicationGroups](#) API den Clustermodus-Parameter als zurück `disabled`.

## AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher

### Note

Der globale Datenspeicher ist derzeit nur für selbst entworfene Cluster verfügbar.

Mithilfe der Global Datastore-Funktion können Sie mit einer vollständig verwalteten, schnellen, zuverlässigen und sicheren Valkey- oder Redis-Cluster-Replikation über Regionen hinweg arbeiten.

OSS AWS Mit dieser Funktion können Sie regionsübergreifende Read Replica-Cluster erstellen, um Lesevorgänge mit niedriger Latenz und regionsübergreifende Disaster Recovery zu ermöglichen.

AWS

In den folgenden Abschnitten finden Sie eine Beschreibung der Arbeit mit globalen Datenspeichern.

Themen

- [Übersicht](#)
- [Voraussetzungen und Einschränkungen](#)
- [Verwendung globaler Datenspeicher \(Konsole\)](#)
- [Verwendung globaler Datenspeicher \(\) CLI](#)

## Übersicht

Jeder globale Datenspeicher ist eine Sammlung von einem oder mehreren Clustern, die zueinander repliziert werden.

Ein globaler Datenspeicher besteht aus folgenden Komponenten:

- Primärer (aktiver) Cluster – Ein primärer Cluster nimmt Schreibvorgänge entgegen, die auf alle Cluster innerhalb des globalen Datenspeichers repliziert werden. Ein primärer Cluster akzeptiert auch Leseanforderungen.
- Sekundärer (passiver) Cluster – Ein sekundärer Cluster nimmt nur Leseanforderungen entgegen und repliziert Datenaktualisierungen von einem primären Cluster. Ein sekundärer Cluster muss sich in einer anderen AWS Region als der primäre Cluster befinden.

Wenn Sie ElastiCache mit Valkey oder Redis einen globalen Datenspeicher erstellen, repliziert diese Ihre Daten automatisch vom primären Cluster auf den sekundären Cluster. Sie wählen die AWS Region aus, in der die Valkey- oder OSS Redis-Daten repliziert werden sollen, und erstellen dann einen sekundären Cluster in dieser Region. AWS ElastiCache richtet dann die automatische, asynchrone Replikation von Daten zwischen den beiden Clustern ein und verwaltet sie.

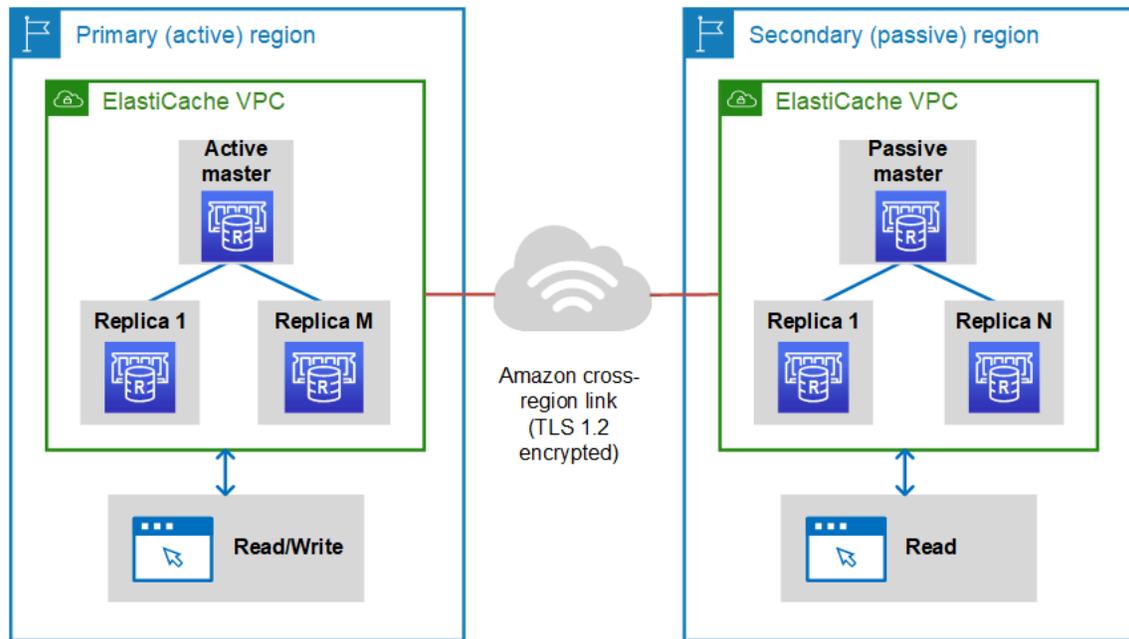
Die Verwendung eines globalen Datenspeichers für Valkey oder Redis OSS bietet die folgenden Vorteile:

- Geolokale Leistung — Indem Sie Remote-Replikat-Cluster in zusätzlichen AWS Regionen einrichten und Ihre Daten zwischen diesen Regionen synchronisieren, können Sie die Latenz beim Datenzugriff in dieser Region reduzieren. AWS Ein globaler Datenspeicher kann dazu

beitragen, die Reaktionsfähigkeit Ihrer Anwendung zu erhöhen, indem er geolokale Lesevorgänge mit niedriger Latenz in allen Regionen bereitstellt. AWS

- Notfallwiederherstellung – Wenn Ihr primärer Cluster in einem globalen Datenspeicher eine Verschlechterung aufweist, können Sie einen sekundären Cluster als Ihren neuen primären Cluster heraufstufen. Sie können dies tun, indem Sie eine Verbindung zu einer beliebigen Region herstellen, die einen sekundären Cluster enthält AWS .

Das folgende Diagramm zeigt, wie globale Datenspeicher funktionieren können.



## Voraussetzungen und Einschränkungen

Bevor Sie die Arbeit mit globalen Datenspeichern beginnen, beachten Sie Folgendes:

- Globale Datenspeicher werden in den folgenden AWS Regionen unterstützt: Asien-Pazifik (Seoul, Tokio, Singapur, Sydney, Mumbai und Osaka), Europa (Frankfurt, Paris, London, Irland und Stockholm), USA Ost (Nord-Virginia und Ohio), USA West (Nordkalifornien und Oregon), Südamerika (São Paulo), AWS GovCloud (US-West und US-Ost), Region Kanada (Mitte), China (Peking und Ningxia)
- Alle Cluster – primär und sekundär – in Ihrem globalen Datenspeicher sollten die gleiche Anzahl von primären Knoten, Knotentyp, Engine-Version und Anzahl von Shards (bei aktiviertem Cluster-Modus) aufweisen. Jeder Cluster in Ihrem globalen Datenspeicher kann über eine andere Anzahl von Read Replicas verfügen, um mit dem lokalen Lesedatenverkehr in diesem Cluster umgehen zu können.

Die Replikation muss aktiviert sein, wenn Sie einen vorhandenen Cluster mit einem einzelnen Knoten verwenden möchten.

- Globale Datenspeicher werden auf Instances mit einer Größe von mindestens einer Größe unterstützt.
- Sie können die Replikation für einen primären Cluster von einer AWS Region zu einem sekundären Cluster in bis zu zwei anderen AWS Regionen einrichten.

#### Note

Eine Ausnahme sind die Regionen China (Peking) und China (Ningxia), wo eine Replikation nur zwischen den beiden Regionen erfolgen kann.

- Sie können mit globalen Datenspeichern nur in VPC Clustern arbeiten. Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC](#). Globale Datenspeicher werden nicht unterstützt, wenn Sie -Classic verwenden. EC2 Weitere Informationen finden Sie unter [EC2-Classic](#) im EC2Amazon-Benutzerhandbuch.

#### Note

Derzeit können Sie keine globalen Datenspeicher in [Verwenden von lokalen Zonen mit ElastiCache](#) verwenden.

- ElastiCache unterstützt kein automatisches Failover von einer AWS Region in eine andere. Bei Bedarf können Sie einen sekundären Cluster manuell heraufstufen. Ein Beispiel finden Sie unter [Heraufstufen des sekundären Clusters zum primären Cluster](#).
- Um eine Bootstrap-Aktion von vorhandenen Daten durchzuführen, verwenden Sie einen vorhandenen Cluster als primären Cluster, um einen globalen Datenspeicher zu erstellen. Das Hinzufügen eines vorhandenen Clusters als sekundärer Cluster wird nicht unterstützt. Beim Hinzufügen des Clusters als sekundärer Cluster werden Daten gelöscht, was zu Datenverlusten führen kann.
- Parameteraktualisierungen werden auf alle Cluster angewendet, wenn Sie eine lokale Parametergruppe eines Clusters ändern, der zu einem globalen Datenspeicher gehört.
- Sie können regionale Cluster sowohl vertikal (Skalierung nach oben und unten) als auch horizontal (Ein- und Ausskalierung) skalieren. Sie können die Cluster skalieren, indem Sie den globalen Datenspeicher ändern. Alle regionalen Cluster in dem globalen Datenspeicher werden dann ohne Unterbrechung skaliert. Weitere Informationen finden Sie unter [Skalierung ElastiCache](#).

- Globale Datenspeicher unterstützen [Verschlüsselung im Ruhezustand, Verschlüsselung bei der Übertragung](#) und. [AUTH](#)
- Globale Datenspeicher unterstützen das Internetprotokoll Version 6 () nicht. IPv6
- Globale Datenspeicher unterstützen Schlüssel. AWS KMS Weitere Informationen finden Sie unter [AWS Schlüsselverwaltungsdienstkonzepte](#) im AWS Key Management Service Entwicklerhandbuch.

#### Note

Globale Datenspeicher unterstützen [Pub/Sub-Messaging](#) mit den folgenden Vorgaben:

- Bei deaktiviertem Cluster-Modus wird Pub/Sub vollständig unterstützt. Ereignisse, die auf dem primären Cluster der primären AWS Region veröffentlicht wurden, werden an sekundäre Regionen weitergegeben. AWS
- Bei aktiviertem Cluster-Modus gilt Folgendes:
  - Bei veröffentlichten Ereignissen, die sich nicht in einem Schlüsselraum befinden, erhalten nur Abonnenten in derselben AWS Region die Ereignisse.
  - Bei veröffentlichten Keyspace-Veranstaltungen erhalten Abonnenten in allen AWS Regionen die Ereignisse.

## Verwendung globaler Datenspeicher (Konsole)

Führen Sie den folgenden aus zwei Schritten bestehenden Vorgang durch, um einen globalen Datenspeicher mit der Konsole zu erstellen:

1. Erstellen Sie einen primären Cluster, entweder mithilfe eines vorhandenen Clusters oder durch Erstellen eines neuen Clusters. Die Engine muss Valkey 7.2 oder höher oder Redis OSS 5.0.6 oder höher sein.
2. Fügen Sie bis zu zwei sekundäre Cluster in verschiedenen AWS Regionen hinzu, wiederum mit Valkey 7.2 oder höher oder der Redis OSS 5.0.6-Engine oder höher.

In den folgenden Verfahren erfahren Sie, wie Sie einen globalen Datenspeicher für Valkey oder Redis erstellen OSS und andere Vorgänge mithilfe der Konsole ausführen. ElastiCache

### Themen

- [Erstellen eines globalen Datenspeichers mit einem vorhandenen Cluster](#)

- [Erstellen eines neuen globalen Datenspeichers mit einem neuen primären Cluster](#)
- [Anzeigen von Details zu einem globalen Datenspeicher](#)
- [Hinzufügen einer Region zu einem globalen Datenspeicher](#)
- [Ändern eines globalen Datenspeichers](#)
- [Heraufstufen des sekundären Clusters zum primären Cluster](#)
- [Entfernen einer Region aus einem globalen Datenspeicher](#)
- [Löschen eines globalen Datenspeichers](#)

## Erstellen eines globalen Datenspeichers mit einem vorhandenen Cluster

In diesem Szenario verwenden Sie einen vorhandenen Cluster, der als primärer Cluster des neuen globalen Datenspeichers dient. Anschließend erstellen Sie einen sekundären, schreibgeschützten Cluster in einer separaten AWS -Region. Dieser sekundäre Cluster erhält automatische und asynchrone Updates vom primären Cluster.

### Important

Der vorhandene Cluster muss eine Engine verwenden, die Valkey 7.2 oder höher oder Redis 5.0.6 oder höher ist. OSS

So erstellen Sie einen globalen Datenspeicher unter Verwendung eines vorhandenen Clusters:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Global Datastores und dann Create Global Datastore aus.
3. Gehen Sie auf der Seite mit den Einstellungen für den primären Cluster wie folgt vor:
  - Geben Sie im Feld Informationen zum globalen Datenspeicher einen Namen für den neuen globalen Datenspeicher ein.
  - (Optional) Geben Sie einen Wert für Description (Beschreibung) ein.
4. Wählen Sie unter Regionaler Cluster die Option Bestehenden regionalen Cluster verwenden aus.
5. Wählen Sie unter Existierender Cluster den vorhandenen Cluster aus, den Sie verwenden möchten.

6. Behalten Sie die folgenden Optionen unverändert bei. Sie sind so vorausgefüllt, dass sie der primären Clusterkonfiguration entsprechen. Sie können sie nicht ändern.

- Engine-Version
- Knotentyp
- Parametergruppe

 Note

ElastiCache generiert automatisch eine neue Parametergruppe aus Werten der angegebenen Parametergruppe und wendet die neue Parametergruppe auf den Cluster an. Verwenden Sie diese neue Parametergruppe, um Parameter in einem globalen Datenspeicher zu ändern. Jede automatisch generierte Parametergruppe ist einem Cluster und daher nur einem globalen Datenspeicher zugeordnet.

- Anzahl der Shards
- Verschlüsselung im Ruhezustand – Aktiviert die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie können einen anderen Verschlüsselungsschlüssel angeben, indem Sie vom Kunden verwalteter AWS KMS Schlüssel und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Vom Kunden verwaltete AWS KMS Schlüssel verwenden](#).

- Verschlüsselung während der Übertragung – Aktiviert die Verschlüsselung von Daten bei der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2 und höher und Redis OSS Engine Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
  - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies weist auf keine Einschränkungen hin.
  - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einem definierten Satz von Benutzern und Berechtigungen für verfügbare Vorgänge aus. Weitere Informationen finden Sie unter [Benutzergruppen mit der Konsole verwalten und CLI](#).

- AUTHStandardbenutzer — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis-Server. OSS Weitere Informationen finden Sie unter. [AUTH](#)
7. (Optional) Aktualisieren Sie bei Bedarf die verbleibenden Einstellungen für den sekundären Cluster. Diese werden mit den gleichen Werten wie beim primären Cluster vorausgefüllt, Sie können sie jedoch aktualisieren, um bestimmte Anforderungen für diesen Cluster zu erfüllen.
- Port
  - Anzahl der Replikate
  - Subnetzgruppe
  - Bevorzugte Availability Zone(s)
  - Sicherheitsgruppen
  - Vom Kunden verwaltet (AWS KMSSchlüssel)
  - AUTHToken
  - Aktivieren automatischer Sicherungen.
  - Aufbewahrungszeitraum für Backups
  - Backup-Fenster
  - Wartungsfenster
  - Thema für die SNS Benachrichtigung
8. Wählen Sie Create (Erstellen) aus. Dadurch wird der Status des globalen Datenspeichers auf Creating (Erstellen) festgelegt. Der Status wechselt zu Modifying (Ändern), nachdem der primäre Cluster dem globalen Datenspeicher zugeordnet wurde und der sekundäre Cluster den Status Associating (Zuordnen) hat.

Nachdem der primäre Cluster und die sekundären Cluster mit dem globalen Datenspeicher verknüpft wurden, ändert sich der Status zu Available (Verfügbar). An dieser Stelle verfügen Sie über einen primären Cluster, der Lese- und Schreibvorgänge akzeptiert, sowie sekundäre Cluster, die vom primären Cluster replizierte Lesevorgänge akzeptieren.

Die Seite wird aktualisiert und gibt nun an, ob ein Cluster Teil eines globalen Datenspeichers ist. Dazu gehören:

- Globaler Datenspeicher – Der Name des globalen Datenspeichers, zu dem der Cluster gehört.
- Globale Datenspeicherrolle – Die Rolle des Clusters, entweder primär oder sekundär.

Sie können bis zu einem zusätzlichen sekundären Cluster in einer anderen AWS Region hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen einer Region zu einem globalen Datenspeicher](#).

Erstellen eines neuen globalen Datenspeichers mit einem neuen primären Cluster

Wenn Sie einen globalen Datenspeicher mit einem neuen Cluster erstellen möchten, gehen Sie folgendermaßen vor.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Global Datastores und dann Create Global Datastore aus.
3. Führen Sie unter Primary cluster settings (Primäre Cluster-Einstellungen) die folgenden Schritte aus:
  - a. Wählen Sie für Cluster mode (Cluster-Modus) Enabled (Aktiviert) oder Disabled (Deaktiviert) aus.
  - b. Geben Sie für Informationen zum globalen Datenspeicher einen Wert für Name ein. ElastiCache verwendet das Suffix, um einen eindeutigen Namen für den globalen Datenspeicher zu generieren. Sie können nach dem globalen Datenspeicher suchen, indem Sie das hier angegebene Suffix verwenden.
  - c. (Optional) Geben Sie einen Wert für die Global Datastore description (Beschreibung des globalen Datenspeichers) ein.
4. Unter Regional cluster (Regionaler Cluster):
  - a. Wählen Sie für Region eine verfügbare Region aus. AWS
  - b. Wählen Sie Create new regional cluster (Neuen regionalen Cluster erstellen) oder Use existing regional cluster (Bestehenden regionalen Cluster verwenden) aus.
  - c. Wenn Sie Create new regional cluster (Neuen regionalen Cluster erstellen), auswählen, geben Sie unter Cluster infos (Cluster-Infos) einen Namen und eine optionale Beschreibung des Clusters ein.
  - d. Wir empfehlen, unter Location (Speicherort) die Voreinstellungen für Multi-AZ und Auto-failover (Automatisches Failover) zu akzeptieren.
5. Unter Cluster settings (Cluster-Einstellungen):
  - a. Wählen Sie für Engine version (Engine-Version) eine verfügbare Version (5.0.6 oder höher).
  - b. Verwenden Sie für Port den Standardport 6379. Wenn es einen Grund gibt, einen anderen Port zu verwenden, geben Sie die betreffende Portnummer ein.

- c. Wählen Sie für die Parameter group (Parametergruppe) eine Parametergruppe aus oder erstellen Sie eine neue Parametergruppe. Parametergruppen steuern die Laufzeitparameter Ihres Clusters. Weitere Informationen zu Parametergruppen finden Sie unter [Valkey- und Redis-Parameter OSS](#) und [Eine ElastiCache Parametergruppe erstellen](#).

 Note

Wenn Sie eine Parametergruppe zum Festlegen der Engine-Konfigurationswerte auswählen, wird diese Parametergruppe auf alle Cluster im globalen Datenspeicher angewendet. Auf der Seite Parameter Groups (Parametergruppen) gibt das Ja/nein-Attribut Global an, ob eine Parametergruppe Teil eines globalen Datenspeichers ist.

- d. Klicken Sie für Node type (Knotentyp) auf den Abwärtspfeil (▼) und wählen Sie im Dialogfeld Change node type (Knotentyp ändern) einen Wert für Instance family (Instance-Familie) für den gewünschten Knotentyp aus. Wählen Sie dann den Knotentyp aus, den Sie für diesen Cluster verwenden möchten, und wählen Sie dann Save (Speichern).

Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).

Wenn Sie einen r6gd-Knotentyp wählen, wird Daten-Tiering automatisch aktiviert. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

- e. Wenn Sie einen Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) erstellen:

Wählen Sie unter Number of replicas (Anzahl der Replikate) die für diesen Cluster gewünschte Anzahl von Replikaten aus.

- f. Wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) erstellen:

- i. Wählen Sie unter Anzahl der Shards die Anzahl der Shards (Partitionen/Knotengruppen) aus, die Sie für diesen Valkey- oder Redis-Cluster (Clustermodus aktiviert) benötigen. OSS

Bei einigen Versionen von Valkey oder Redis OSS (Clustermodus aktiviert) können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern:

- Redis OSS 3.2.10 und höher — Wenn auf Ihrem Cluster Redis OSS 3.2.10 oder spätere Versionen ausgeführt werden, können Sie die Anzahl der Shards in Ihrem

Cluster dynamisch ändern. Weitere Informationen finden Sie unter [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#).

- Andere OSS Redis-Versionen — Wenn auf Ihrem Cluster eine Version von Redis vor Version 3.2.10 ausgeführt wird, gibt es einen anderen Ansatz OSS. Um die Anzahl der Shards in Ihrem Cluster in diesem Fall zu ändern, erstellen Sie einen neuen Cluster mit der neuen Anzahl von Shards. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
- ii. Wählen Sie für Replicas per shard (Replikate pro Shard) die Anzahl der Read Replica-Knoten aus, die sich in jedem Shard befinden sollen.

Die folgenden Einschränkungen gelten für Valkey oder Redis OSS (Clustermodus aktiviert).

- Wenn Sie Multi-AZ aktiviert haben, stellen Sie sicher, dass mindestens ein Replikat pro Shard vorhanden ist.
  - Die Anzahl der Replikate ist für jeden Shard gleich, wenn der Cluster mithilfe der Konsole erstellt wird.
  - Die Anzahl der Lesereplikate pro Shard ist ein fester Wert, der nicht geändert werden kann. Wenn Sie feststellen, dass Sie mehr oder weniger Replikate pro Shard (API/CLI: Knotengruppe) benötigen, müssen Sie einen neuen Cluster mit der neuen Anzahl von Replikaten erstellen. Weitere Informationen finden Sie unter [Tutorial: Seeding eines neuen, selbst entworfenen Clusters mit einem extern erstellten Backup](#).
6. Wählen Sie unter Einstellungen für Subnetzgruppen das Subnetz aus, das Sie auf diesen Cluster anwenden möchten. ElastiCache stellt eine IPv4 Standard-Subnetzgruppe bereit, oder Sie können wählen, ob Sie eine neue erstellen möchten. Denn IPv6 Sie müssen eine Subnetzgruppe mit einem IPv6 CIDR Block erstellen. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Weitere Informationen finden Sie unter [Erstellen Sie ein Subnetz in Ihrem VPC](#).

7. Für Availability zone placements (Availability-Zone-Platzierungen) haben Sie zwei Optionen:
- Keine Präferenz — ElastiCache wählt die Availability Zone.
  - Availability Zones angeben – Sie geben die Availability Zone für jeden Cluster an.

Wenn Sie die Availability Zones angeben, wählen Sie für jeden Cluster in jedem Shard die Availability Zone aus der Liste aus.

Weitere Informationen finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

|             | 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 |

Angeben von Keyspaces und Availability Zones

8. Wählen Sie Next (Weiter).
9. Unter Erweiterte Valkey- und Redis-Einstellungen OSS
  - Für Security (Sicherheit):
    - i. Zur Verschlüsselung Ihrer Daten haben Sie die folgenden Optionen:
      - Verschlüsselung im Ruhezustand – Ermöglicht die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

**Note**

Sie haben die Möglichkeit, einen anderen Verschlüsselungsschlüssel anzugeben, indem Sie vom Kunden verwalteter AWS KMS Schlüssel und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Vom Kunden verwaltete Schlüssel verwenden von AWS KMS](#).

- Verschlüsselung während der Übertragung – Ermöglicht die Verschlüsselung von Daten während der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2 und höher und Redis OSS Engine Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
  - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
  - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern aus, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Benutzergruppen mit der Konsole verwalten und CLI](#).
  - AUTHStandardbenutzer — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis-Server. OSS Weitere Informationen finden Sie unter. [AUTH](#)
  - AUTH— Ein Authentifizierungsmechanismus für einen Valkey- oder Redis-ServerOSS. Weitere Informationen finden Sie unter. [AUTH](#)

 Note

Für OSS Redis-Versionen zwischen 3.2.6 und höher, mit Ausnahme von Version 3.2.10, AUTH ist dies die einzige Option.

- ii. Wählen Sie für Security groups (Sicherheitsgruppen) die gewünschten Sicherheitsgruppen für diesen Cluster aus. Eine security group (Sicherheitsgruppe) fungiert als Firewall, um den Netzwerkzugriff auf Ihren Cluster zu steuern. Sie können die Standardsicherheitsgruppe für Ihre verwenden VPC oder eine neue erstellen.

Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Sicherheitsgruppen für Sie VPC](#) im VPCAmazon-Benutzerhandbuch.

10. Wenn Sie regelmäßig geplante automatische Sicherungen möchten, aktivieren Sie Enable automatic backups und geben Sie ein, wie viele Tage lang jede automatische Sicherung beibehalten werden soll, bevor sie automatisch gelöscht wird. Wenn Sie keine regelmäßig geplanten automatischen Sicherungen möchten, deaktivieren Sie das Kontrollkästchen Enable automatic backups. In beiden Fällen haben Sie jederzeit die Option, manuelle Sicherungen zu erstellen.

Weitere Informationen zur Sicherung und Wiederherstellung finden Sie unter [Snapshot und Wiederherstellung](#).

11. (Optional) Geben Sie ein Wartungsfenster an. Das Wartungsfenster ist der Zeitraum, der in der Regel eine Stunde lang ist, jede Woche, zu der ElastiCache die Systemwartung für Ihren Cluster geplant wird. Sie können ElastiCache den Tag und die Uhrzeit für Ihr Wartungsfenster wählen lassen (keine Präferenz), oder Sie können Tag, Uhrzeit und Dauer selbst wählen (Wartungsfenster angeben). Treffen Sie bei Wahl von Specify maintenance window eine Auswahl in den Listen Start day, Start time und Duration (in Stunden) für Ihr Wartungsfenster. Alle Zeiten sind UCT Zeiten.

Weitere Informationen finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

12. (Optional) Für Protokolle:
  - Wählen Sie unter Protokollformat entweder Text oder aus JSON.
  - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
  - Wählen Sie unter Protokollziel entweder Neu erstellen und geben Sie entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und wählen Sie dann entweder Ihren CloudWatch Logs-Log-Gruppennamen oder Ihren Firehose-Streamnamen aus.
13. Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen unter Tags zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#).
14. Überprüfen Sie alle Ihre Einträge und ausgewählten Optionen und machen Sie dann evtl. erforderliche Korrekturen. Wählen Sie danach Next (Weiter) aus.
15. Nachdem Sie den Cluster in den vorherigen Schritten konfiguriert haben, konfigurieren Sie nun die Details des sekundären Clusters.
16. Wählen Sie unter Regionaler Cluster die AWS Region aus, in der sich der Cluster befindet.
17. Geben Sie unter Cluster info (Cluster-Info) einen Namen und eine optionale Beschreibung des Clusters ein.
18. Die folgenden Optionen sind für die Konfiguration des primären Clusters vorausgefüllt und können nicht geändert werden:
  - Ort
  - Engine-Version

- Instance-Typ
- Knotentyp
- Anzahl der Shards
- Parametergruppe

 Note

ElastiCache generiert automatisch eine neue Parametergruppe aus Werten der angegebenen Parametergruppe und wendet die neue Parametergruppe auf den Cluster an. Verwenden Sie diese neue Parametergruppe, um Parameter in einem globalen Datenspeicher zu ändern. Jede automatisch generierte Parametergruppe ist einem Cluster und daher nur einem globalen Datenspeicher zugeordnet.

- Verschlüsselung im Ruhezustand – Aktiviert die Verschlüsselung von Daten, die auf der Festplatte gespeichert sind. Weitere Informationen finden Sie unter [Verschlüsselung im Ruhezustand](#).

 Note

Sie können einen anderen Verschlüsselungsschlüssel angeben, indem Sie vom Kunden verwalteter AWS KMS Schlüssel und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Vom Kunden verwaltete AWS KMS Schlüssel verwenden](#).

- Verschlüsselung während der Übertragung – Aktiviert die Verschlüsselung von Daten bei der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Valkey 7.2 und höher und Redis OSS Engine Version 6.4 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Zugriffskontrolloptionen anzugeben:
  - Keine Zugriffskontrolle – Dies ist die Standardeinstellung. Dies bedeutet, dass es keine Einschränkungen für den Benutzerzugang zum Cluster gibt.
  - Zugriffskontrollliste für Benutzergruppen – Wählen Sie eine Benutzergruppe mit einer bestimmten Anzahl von Benutzern, die auf den Cluster zugreifen können. Weitere Informationen finden Sie unter [Benutzergruppen mit der Konsole verwalten und CLI](#).
  - AUTHStandardbenutzer — Ein Authentifizierungsmechanismus für einen Valkey- oder Redis-Server. OSS Weitere Informationen finden Sie unter [AUTH](#)

 Note

Für OSS Redis-Versionen zwischen 4.0.2, als Verschlüsselung bei der Übertragung erstmals unterstützt wurde, und 6.0.4 AUTH ist dies die einzige Option.

Die verbleibenden Einstellungen für den sekundären Cluster werden mit den gleichen Werten wie der primäre Cluster ausgefüllt, aber Folgendes kann aktualisiert werden, um bestimmte Anforderungen für diesen Cluster zu erfüllen.

- Port
  - Anzahl der Replikate
  - Subnetzgruppe
  - Bevorzugte Availability Zone(s)
  - Sicherheitsgruppen
  - Vom Kunden verwaltet (Schlüssel)AWS KMS
  - AUTHToken
  - Aktivieren automatischer Sicherungen.
  - Aufbewahrungszeitraum für Backups
  - Backup-Fenster
  - Wartungsfenster
  - Thema für die SNS Benachrichtigung
19. Wählen Sie Create (Erstellen) aus. Dadurch wird der Status des globalen Datenspeichers auf Creating (Erstellen) gesetzt. Nachdem der primäre Cluster und die sekundären Cluster mit dem globalen Datenspeicher verknüpft wurden, ändert sich der Status zu Available (Verfügbar). Sie verfügen über einen primären Cluster, der Lese- und Schreibvorgänge akzeptiert, und einen sekundären Cluster, der Lesevorgänge akzeptiert, die vom primären Cluster repliziert werden.

Die Seite wurde außerdem aktualisiert und gibt nun an, ob ein Cluster Teil eines globalen Datenspeichers ist. Dazu gehören unter anderem die folgenden Informationen:

- Globaler Datenspeicher – Der Name des globalen Datenspeichers, zu dem der Cluster gehört.
- Globale Datenspeicherrolle – Die Rolle des Clusters, entweder primär oder sekundär.

Sie können bis zu einem zusätzlichen sekundären Cluster in einer anderen AWS Region hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen einer Region zu einem globalen Datenspeicher](#).

## Anzeigen von Details zu einem globalen Datenspeicher

Sie können die Details vorhandener globaler Datenspeicher anzeigen und sie auch auf der Seite Globale Datenspeicher ändern.

So zeigen Sie Details zu einem globalen Datenspeicher an:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Globale Datenspeicher und dann einen verfügbaren globalen Datenspeicher aus.

Anschließend können Sie die folgenden Eigenschaften des globalen Datenspeichers untersuchen:

- Global Datastore Name (Name des globalen Datenspeichers): Der Name des globalen Datenspeichers
- Description (Beschreibung): Eine Beschreibung des globalen Datenspeichers
- Status: Folgende Optionen stehen zur Verfügung:
  - Erstellen
  - Ändern
  - Verfügbar
  - Löschen
  - Nur Primär - Dieser Status gibt an, dass der globale Datenspeicher nur einen primären Cluster enthält. Entweder wurden alle sekundären Cluster gelöscht oder nicht erfolgreich erstellt.
- Cluster Mode (Cluster-Modus): Aktiviert oder deaktiviert.
- Engine-Version: Die Valkey- oder OSS Redis-Engine-Version, auf der der globale Datenspeicher ausgeführt wird
- Instance Node Type (Instance-Knotentyp): Der für den globalen Datenspeicher verwendete Knotentyp
- Encryption at-rest (Verschlüsselung im Ruhezustand): Aktiviert oder deaktiviert
- Encryption in-transit (Verschlüsselung während der Übertragung): Aktiviert oder deaktiviert
- AUTH: Entweder aktiviert oder deaktiviert

Sie können die folgenden Änderungen an dem globalen Datenspeicher vornehmen:

- [Hinzufügen einer Region zu einem globalen Datenspeicher](#)
- [Entfernen einer Region aus einem globalen Datenspeicher](#)
- [Heraufstufen des sekundären Clusters zum primären Cluster](#)
- [Ändern eines globalen Datenspeichers](#)

Auf der Seite „Global Datastore (Globaler Datenspeicher)“ werden auch die einzelnen Cluster aufgelistet, aus denen der globale Datenspeicher besteht, sowie die folgenden Eigenschaften für jeden davon:

- Region — Die AWS Region, in der der Cluster gespeichert ist
- Role (Rolle) - Entweder primär oder sekundär
- Cluster name (Cluster-Name) - Der Name des Clusters
- Status - Zu den Optionen gehören:
  - Associating (Zuordnen) - Der Cluster wird gerade dem globalen Datenspeicher zugeordnet
  - Associated (Zugeordnet) - Der Cluster ist dem globalen Datenspeicher zugeordnet
  - Disassociating (Trennen) - Der Vorgang des Entfernens eines sekundären Clusters aus dem globalen Datenspeicher unter Verwendung des Namens des globalen Datenspeichers. Danach erhält der sekundäre Cluster keine Updates mehr vom primären Cluster, sondern er verbleibt als eigenständiger Cluster in dieser AWS Region.
  - Disassociated (Getrennt) - Der sekundäre Cluster wurde aus dem globalen Datenspeicher entfernt und ist jetzt ein eigenständiger Cluster in seiner AWS -Region.
- Global Datastore Replica Lag — Zeigt einen Wert pro sekundärer AWS Region im globalen Datenspeicher an. Dies ist die Verzögerung zwischen dem Primärknoten der sekundären Region und dem Primärknoten der primären Region. Bei aktiviertem Valkey oder Redis OSS im Clustermodus gibt die Verzögerung die maximale Verzögerung in Sekunden zwischen den Shards an.

## Hinzufügen einer Region zu einem globalen Datenspeicher

Sie können einem vorhandenen globalen Datenspeicher bis zu eine zusätzliche AWS Region hinzufügen. In diesem Szenario erstellen Sie einen schreibgeschützten Cluster in einer separaten AWS Region, der automatische und asynchrone Updates vom primären Cluster empfängt.

## Um eine AWS Region zu einem globalen Datenspeicher hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Globale Datenspeicher und dann einen vorhandenen globalen Datenspeicher aus.
3. Wählen Sie Regionalen Cluster hinzufügen und wählen Sie die AWS Region aus, in der sich der sekundäre Cluster befinden soll.
4. Geben Sie unter Clusterinformationen einen Wert für Name und optional für Beschreibung für den Cluster ein.
5. Behalten Sie die folgenden Optionen unverändert bei. Sie sind bereits so ausgefüllt, dass sie der primären Clusterkonfiguration entsprechen, und Sie können sie nicht ändern.
  - Engine-Version
  - Instance-Typ
  - Knotentyp
  - Anzahl der Shards
  - Parametergruppe

### Note

ElastiCache generiert automatisch eine neue Parametergruppe aus Werten der angegebenen Parametergruppe und wendet die neue Parametergruppe auf den Cluster an. Verwenden Sie diese neue Parametergruppe, um Parameter in einem globalen Datenspeicher zu ändern. Jede automatisch generierte Parametergruppe ist einem Cluster und daher nur einem globalen Datenspeicher zugeordnet.

- Verschlüsselung im Ruhezustand

### Note

Sie können einen anderen Verschlüsselungsschlüssel angeben, indem Sie vom Kunden verwalteter AWS KMS Schlüssel und dann den Schlüssel auswählen.

- Verschlüsselung während der Übertragung
- AUTH

6. (Optional) Aktualisieren Sie die verbleibenden Einstellungen für den sekundären Cluster. Diese werden mit den gleichen Werten wie beim primären Cluster vorausgefüllt, Sie können sie jedoch aktualisieren, um bestimmte Anforderungen für diesen Cluster zu erfüllen.
  - Port
  - Anzahl der Replikate
  - Subnetzgruppe
  - Bevorzugte Availability Zone(s)
  - Sicherheitsgruppen
  - Vom Kunden verwalteter AWS KMS Schlüssel)
  - AUTHToken
  - Aktivieren automatischer Sicherungen.
  - Aufbewahrungszeitraum für Backups
  - Backup-Fenster
  - Wartungsfenster
  - Thema für die SNS Benachrichtigung
7. Wählen Sie Hinzufügen aus.

## Ändern eines globalen Datenspeichers

Sie können die Eigenschaften regionaler Cluster ändern. In einem globalen Datenspeicher kann nur jeweils eine Änderungsoperation ausgeführt werden, mit Ausnahme der Heraufstufung eines sekundären Clusters zu einem primären Cluster. Weitere Informationen finden Sie unter [Heraufstufen des sekundären Clusters zum primären Cluster](#).

So ändern Sie einen globalen Datenspeicher:

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Global Datastores und dann für Global Datastore Name einen globalen Datenspeicher aus.
3. Wählen Sie Modify (Ändern), und wählen Sie unter den folgenden Optionen:
  - Modify description (Beschreibung ändern) – Aktualisieren der Beschreibung des globalen **Datenspeichers**

- Engine-Version ändern — Nur Valkey 7.2 und höher oder Redis OSS Engine-Version 5.0.6 und höher sind verfügbar.
- Modify node type (Knotentyp ändern) – Vertikale (Aufwärts- und Abwärts-) und horizontale (Ein- und Aus-) Skalierung regionaler Cluster. Zu den Optionen gehören die R5- und M5-Knotenfamilien. Weitere Informationen zu Knotentypen finden Sie unter [Unterstützte Knotentypen](#).
- Modify Automatic Failover (Ändern des automatischen Failovers) – Aktivieren oder deaktivieren Sie das automatische Failover. Wenn Sie Failover aktivieren und Primärknoten in regionalen Clustern unerwartet heruntergefahren werden, erfolgt ein Failover ElastiCache zu einem der regionalen Replikate. Weitere Informationen finden Sie im Abschnitt [Automatisches Failover](#).

Für Valkey- oder OSS Redis-Cluster mit aktiviertem Clustermodus:

- Add shards (Shards hinzufügen) – Geben Sie die Anzahl der hinzuzufügenden Shards ein, und geben Sie eine oder mehrere Availability Zones an.
- Shards löschen — Wählen Sie Shards aus, die in jeder Region gelöscht werden sollen. AWS
- Rebalance shards (Shards neu ausbalancieren) – Balancieren Sie die Slot-Verteilung neu aus, um eine gleichmäßige Verteilung über die in dem Cluster vorhandenen Shards zu gewährleisten.

Um die Parameter eines globalen Datenspeichers zu ändern, ändern Sie die Parametergruppe eines beliebigen Mitgliedsclusters für den globalen Datenspeicher. ElastiCache wendet diese Änderung automatisch auf alle Cluster innerhalb dieses globalen Datenspeichers an. Um die Parametergruppe dieses Clusters zu ändern, verwenden Sie die Valkey- oder OSS Redis-Konsole oder den Vorgang. [ModifyCacheClusterAPI](#) Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#). Wenn Sie die Parametergruppe eines Clusters ändern, der in einem globalen Datenspeicher enthalten ist, wird sie auf alle Cluster innerhalb dieses globalen Datenspeichers angewendet.

Verwenden Sie den Vorgang, um eine gesamte Parametergruppe oder bestimmte Parameter zurückzusetzen. [ResetCacheParameterGroupAPI](#)

Heraufstufen des sekundären Clusters zum primären Cluster

Wenn der primäre Cluster oder die AWS Region nicht mehr verfügbar ist oder Leistungsprobleme auftreten, können Sie einen sekundären Cluster zum primären Cluster heraufstufen. Die

Heraufstufung ist jederzeit erlaubt, auch wenn gleichzeitig andere Änderungen vorgenommen werden. Sie können auch mehrere Heraufstufungen gleichzeitig durchführen, und der globale Datenspeicher wird schließlich zu einem primären Cluster aufgelöst. Wenn Sie mehrere sekundäre Cluster gleichzeitig hochstufen, kann ElastiCache mit Valkey oder Redis OSS nicht garantiert werden, welcher letztendlich zum primären Cluster wird.

So stufen Sie einen sekundären Cluster zu einem primären Cluster herauf:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Global Datastores aus.
3. Wählen Sie den Namen des globalen Datenspeichers aus, um die Details anzuzeigen.
4. Wählen Sie den Secondary (sekundären) Cluster.
5. Wählen Sie Auf Primär heraufstufen.

Sie werden dann mit der folgenden Warnung aufgefordert, Ihre Entscheidung zu bestätigen:  
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. Wählen Sie Confirm (Bestätigen), wenn Sie die Heraufstufung fortsetzen möchten, oder Cancel (Abbrechen), wenn Sie dies nicht möchten.

Wenn Sie sich für die Bestätigung entscheiden, wechselt Ihr globaler Datenspeicher in den Status Modifying (Ändern) und ist erst verfügbar, wenn die Heraufstufung abgeschlossen ist.

### Entfernen einer Region aus einem globalen Datenspeicher

Mithilfe des folgenden Verfahrens können Sie eine AWS Region aus einem globalen Datenspeicher entfernen.

## Um eine AWS Region aus einem globalen Datenspeicher zu entfernen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Global Datastores aus.
3. Wählen Sie einen globalen Datenspeicher aus.
4. Wählen Sie die Region aus, die Sie entfernen möchten.
5. Wählen Sie Remove region (Region entfernen).

### Note

Diese Option ist nur für sekundäre Cluster verfügbar.

Sie werden dann mit der folgenden Warnung aufgefordert, Ihre Entscheidung zu bestätigen:  
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. Wählen Sie Confirm (Bestätigen), wenn Sie die Heraufstufung fortsetzen möchten, oder Cancel (Abbrechen), wenn Sie dies nicht möchten.

Wenn Sie „Bestätigen“ wählen, wird die AWS Region entfernt und der sekundäre Cluster erhält keine Replikationsupdates mehr.

## Löschen eines globalen Datenspeichers

Um einen globalen Datenspeicher zu löschen, entfernen Sie zuerst alle sekundären Cluster. Weitere Informationen finden Sie unter [Entfernen einer Region aus einem globalen Datenspeicher](#). Dadurch bleibt der globale Datenspeicher im Status primary-only.

So löschen Sie einen globalen Datenspeicher:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Global Datastores aus.

3. Wählen Sie unter Global Datastore Name (Name des globalen Datenspeichers) den globalen Datenspeicher, den Sie löschen möchten, und dann Delete (Löschen).

Sie werden dann mit der folgenden Warnung aufgefordert, Ihre Entscheidung zu bestätigen: Are you sure you want to delete this Global Datastore?

4. Wählen Sie Löschen.

Der globale Datenspeicher wechselt zum Status Deleting (Löschen).

## Verwendung globaler Datenspeicher ( ) CLI

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um mehrere AWS Dienste von der Befehlszeile aus zu steuern und sie mithilfe von Skripten zu automatisieren. Sie können das AWS CLI für (einmalige) Ad-hoc-Operationen verwenden.

### Herunterladen und Konfigurieren der AWS CLI

Das AWS CLI läuft unter Windows, MacOS oder Linux. Gehen Sie folgendermaßen vor, um sie herunterzuladen und zu konfigurieren.

Um das herunterzuladen, zu installieren und zu konfigurieren CLI

1. Laden Sie die Webseite AWS CLI auf der [AWS Befehlszeilenschnittstelle](#) herunter.
2. Folgen Sie den Anweisungen zur Installation AWS CLI und Konfiguration von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

### Verwenden von AWS CLI mit globalen Datenspeichern

Verwenden Sie die folgenden CLI Operationen, um mit globalen Datenspeichern zu arbeiten:

- [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 wendet ElastiCache automatisch ein Präfix auf die globale Datenspeicher-ID an, wenn sie erstellt wird. Jede AWS Region hat ihr eigenes Präfix. Beispielsweise beginnt eine globale

Datenspeicher-ID, die in der Region USA West (Nordkalifornien) erstellt wurde, mit „virxk“ zusammen mit dem von Ihnen angegebenen Suffixnamen. Das Suffix in Kombination mit dem automatisch generierten Präfix garantiert die Eindeutigkeit des globalen Datenspeichernamens über mehrere Regionen hinweg.

In der folgenden Tabelle sind jede AWS Region und ihr globales Datenspeicher-ID-Präfix aufgeführt.

| Regionsname/Region                             | Präfix |
|------------------------------------------------|--------|
| Region USA Ost (Ohio)<br>us-east-2             | fpkhr  |
| Region USA Ost (Nord-Virginia)<br>us-east-1    | ldgnf  |
| Region US West (N. California)<br>us-west-1    | virxk  |
| Region USA West (Oregon)<br>us-west-2          | sgau   |
| Region Kanada (Zentral)<br>ca-central-1        | bxodz  |
| Region Asien-Pazifik (Mumbai)<br>ap-south-1    | erpgt  |
| Region Asien-Pazifik (Tokio)<br>ap-northeast-1 | qusw   |

| Regionsname/Region                                | Präfix |
|---------------------------------------------------|--------|
| Region Asien-Pazifik (Seoul)<br>ap-northeast-2    | lfqnh  |
| Region Asien-Pazifik (Osaka)<br>ap-northeast-3    | nlapn  |
| Region Asien-Pazifik (Singapur)<br>ap-southeast-1 | v1qxn  |
| Region Asien-Pazifik (Sydney)<br>ap-southeast-2   | vbgxd  |
| Region Europa (Frankfurt)<br>eu-central-1         | iudkw  |
| Region Europa (Irland)<br>eu-west-1               | gxeiz  |
| Region Europa (London)<br>eu-west-2               | okuqm  |
| Region Europa (Paris)<br>eu-west-3                | fgjhi  |
| Region Südamerika (São Paulo)<br>sa-east-1        | juxlw  |
| Region China (Peking)<br>cn-north-1               | emvgo  |

| Regionsname/Region                           | Präfix |
|----------------------------------------------|--------|
| Region China (Ningxia)<br>cn-northwest-1     | ckbem  |
| Region Asien-Pazifik (Hongkong)<br>ap-east-1 | knjmp  |
| AWS GovCloud (USA West)<br>us-gov-west-1     | sgwui  |

- [create-replication-group](#)— Verwenden Sie diesen Vorgang, um sekundäre Cluster für einen globalen Datenspeicher zu erstellen, indem Sie den Namen des globalen Datenspeichers für den Parameter angeben. `--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
```

Wenn Sie diesen Vorgang aufrufen und einen `--global-replication-group-id` Wert übergeben, ElastiCache werden die Werte für die folgenden Parameter aus der primären Replikationsgruppe der globalen Replikationsgruppe abgeleitet. Übergeben Sie keine Werte für diese Parameter:

```
"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
```

## Engine-übergreifendes Upgrade von Redis auf Valkey für OSS ElastiCache GlobalDataStore

Sie können eine bestehende OSS globale Redis-Replikationsgruppe mithilfe der Konsole auf die Valkey-Engine aktualisieren, oder. API CLI

Wenn Sie über eine bestehende OSS globale Redis-Replikationsgruppe verfügen, können Sie ein Upgrade auf Valkey durchführen, indem Sie die neue Engine und die Engine-Version mit angeben.  
modify-global-replication-group API

Für Linux, macOS oder Unix:

```
aws elasticache modify-global-replication-group \
 --global-replication-group-id myGlobalReplGroup \
 --engine valkey \
 --apply-immediately \
 --engine-version 7.2
```

Für Windows:

```
aws elasticache modify-global-replication-group ^\
 --global-replication-group-id myGlobalReplGroup ^\
 --engine valkey ^\
 --apply-immediately ^\
 --engine-version 7.2
```

Wenn Sie eine benutzerdefinierte Cache-Parametergruppe auf die bestehende OSS globale Redis-Replikationsgruppe angewendet haben, die Sie aktualisieren möchten, müssen Sie in der Anfrage auch eine benutzerdefinierte Valkey-Cache-Parametergruppe übergeben. Die benutzerdefinierte Valkey-Eingabeparametergruppe muss dieselben OSS statischen Redis-Parameterwerte haben wie die vorhandene benutzerdefinierte OSS Redis-Parametergruppe.

Für Linux, macOS oder 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
```

Für 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-Replikationsgruppe](#)

```
aws elasticache rebalance-shards-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name
```

Verwenden Sie die Hilfe, um alle verfügbaren Befehle für ElastiCache Valkey oder Redis aufzulisten. OSS

```
aws elasticache help
```

Sie können die Hilfe auch verwenden, um einen bestimmten Befehl zu beschreiben und mehr über seine Verwendung zu erfahren:

```
aws elasticache create-global-replication-group help
```

## Hohe Verfügbarkeit mit Replikationsgruppen

Amazon ElastiCache Valkey- und OSS Redis-Cluster mit einem Knoten sind In-Memory-Entitäten mit eingeschränkten Datenschutzdiensten (). AOF Sollte Ihr Cluster aus irgendeinem Grund ausfallen, verlieren Sie alle Daten des Clusters. Wenn Sie jedoch eine Valkey- oder OSS Rediss-Engine ausführen, können Sie 2 bis 6 Knoten zu einem Cluster mit Replikaten gruppieren, wobei 1 bis 5 schreibgeschützte Knoten replizierte Daten des einzelnen primären Lese-/Schreibknotens der Gruppe enthalten. Wenn aus irgendeinem Grund ein Knoten in diesem Szenario ausfällt, verlieren Sie nicht alle Daten, da sie auf einem oder mehreren Knoten repliziert sind. Aufgrund der Replikationslatenz gehen möglicherweise einige Daten verloren, wenn der primäre Lese/Schreib-Knoten ausfällt.

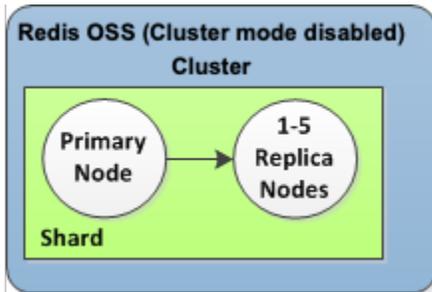
Wie in der folgenden Abbildung zu sehen ist, ist die Replikationsstruktur in einem Shard (imAPI/ als Knotengruppe bezeichnetCLI) enthalten, der in einem Valkey- oder Redis-Cluster enthalten ist. OSS Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) haben immer einen Shard. Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) können bis zu 500 Shards enthalten, wobei die Clusterdaten auf die Shards verteilt sind. Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

Das Knoten- oder Shard-Limit kann mit Valkey und mit der Redis-Engine-Version 5.0.6 oder höher auf maximal 500 pro Cluster erhöht werden. OSS Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze

gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.



Der Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) hat einen Shard und 0 bis 5 Replikatknoten

Wenn für den Cluster mit Replikaten Multi-AZ aktiviert ist, wird bei einem Ausfall des primären Knotens ein automatisches Failover auf eine Read Replica durchgeführt. Da die Daten auf den Replikatknoten asynchron aktualisiert werden, kann die Latenz bei der Aktualisierung der Replikatknoten zu geringfügigem Datenverlust führen. Weitere Informationen finden Sie unter [Minimierung von Fehlern beim Ausführen von Valkey oder Redis OSS](#).

## Themen

- [Valkey- und OSS Redis-Replikation verstehen](#)
- [Replikation: Valkey- und OSS Redis-Clustermodus deaktiviert oder aktiviert](#)
- [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Eine Valkey- oder OSS Redis-Replikationsgruppe erstellen](#)
- [Anzeigen der Details einer Replikationsgruppe](#)
- [Suchen von Endpunkten von Replikationsgruppen](#)
- [Ändern einer Replikationsgruppe](#)
- [Löschen einer Replikationsgruppe](#)
- [Ändern der Anzahl von Replikaten](#)

- [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey- oder Redis-Replikationsgruppen OSS \(Clustermodus deaktiviert\)](#)

## Valkey- und OSS Redis-Replikation verstehen

Redis OSS implementiert die Replikation auf zwei Arten:

- Mit einem einzigen Shard, der alle Clusterdaten in jedem Knoten enthält — Valkey oder Redis (Clustermodus deaktiviert) OSS
- Mit Daten, die auf bis zu 500 Shards partitioniert sind — Valkey oder Redis (Clustermodus aktiviert) OSS

Jeder Shard in einer Replikationsgruppe verfügt über einen einzelnen primären Lese-/Schreibknoten und bis zu 5 schreibgeschützte Replikatknoten. Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die OSS Redis-Engine-Version 5.0.6 oder höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

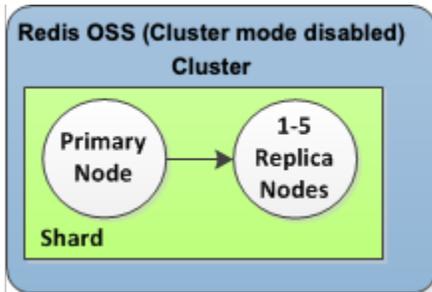
Themen

- [Valkey oder OSS Redis \(Clustermodus deaktiviert\)](#)
- [Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)

Valkey oder OSS Redis (Clustermodus deaktiviert)

Ein Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) hat einen einzelnen Shard, in dem sich eine Sammlung von Knoten befindet: ein primärer Lese-/Schreibknoten und bis zu

fünf sekundäre, schreibgeschützte Replikatknoten. Jede Read Replica speichert eine Kopie der Daten aus dem primären Knoten des Clusters. Mittels asynchroner Replikationsmechanismen werden die Read Replicas mit dem primären Knoten synchron gehalten. Anwendungen können aus jedem Knoten im Cluster lesen. Anwendungen können nur in den primären Knoten schreiben. Read Replicas verbessern den Lesedurchsatz und schützen vor Datenverlust im Falle eines Knotenausfalls.



Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem einzigen Shard und Replikatknoten

Sie können Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit Replikatknoten verwenden, um Ihre Lösung für leseintensive Anwendungen oder für ElastiCache die Unterstützung einer großen Anzahl von Clients zu skalieren, die gleichzeitig aus demselben Cluster lesen.

Alle Knoten in einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) müssen sich in derselben Region befinden.

Wenn Sie eine Read Replica zu einem Cluster hinzufügen, werden alle Daten vom primären Knoten auf den neuen Knoten kopiert. Ab diesem Punkt werden Änderungen jedes Mal, wenn Daten in den primären Knoten geschrieben werden, synchron auf alle Read Replicas übertragen.

Um die Fehlertoleranz zu verbessern und Ausfallzeiten beim Schreiben zu reduzieren, aktivieren Sie Multi-AZ mit automatischem Failover für Ihren Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit Replikaten. Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten durch die Verwendung von ElastiCache von Multi-AZ mit Valkey und Redis OSS](#).

Sie können die Rollen der Knoten innerhalb des Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) ändern, wobei der primäre Knoten und eines der Replikate die Rollen austauschen. Sie könnten sich aus Gründen der Leistungsoptimierung hierfür entscheiden. Bei einer Webanwendung mit der intensivsten Schreibaktivität könnten Sie z. B. den Knoten mit der niedrigsten Netzwerklatenz wählen. Weitere Informationen finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey- oder Redis-Replikationsgruppen OSS \(Clustermodus deaktiviert\)](#).

## Valkey oder Redis OSS (Clustermodus aktiviert)

Ein Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) besteht aus 1 bis 500 Shards (API/CLI: Knotengruppen). Jeder Shard verfügt über einen primären Knoten und bis zu fünf schreibgeschützte Replikatknoten. Die Konfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

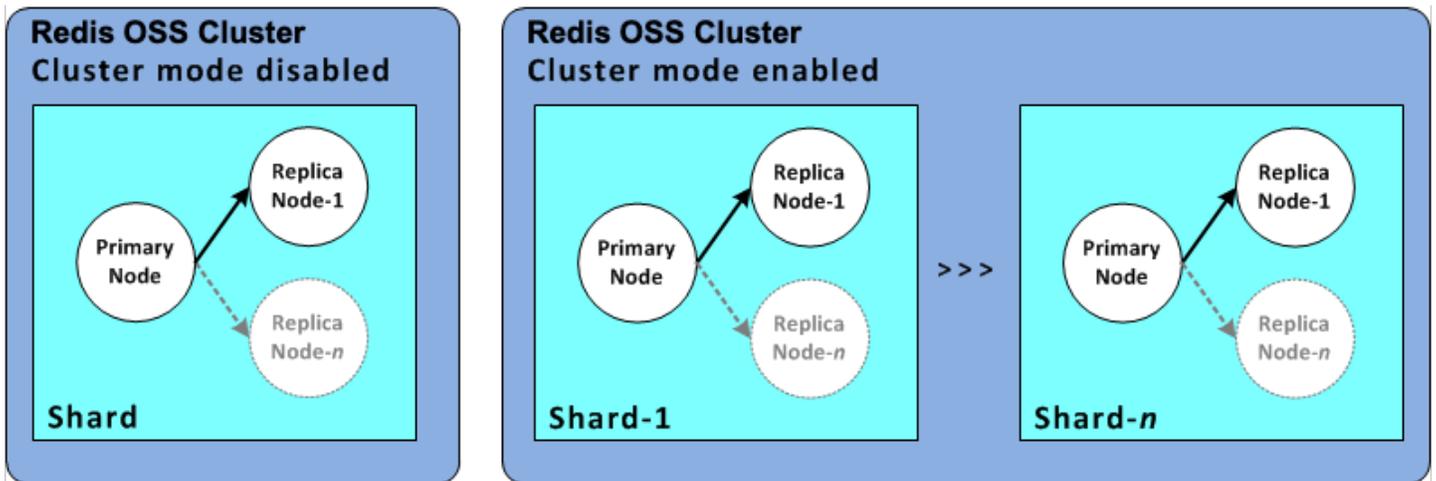
Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Engine-Version Valkey 7.2 und höher oder Redis 5.0.6 und höher ist. OSS Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

Jede Read Replica in einem Shard speichert eine Kopie der Daten aus dem primären Knoten des Shards. Mittels asynchroner Replikationsmechanismen werden die Read Replicas mit dem primären Knoten synchron gehalten. Anwendungen können aus jedem Knoten im Cluster lesen. Anwendungen können nur in die primären Knoten schreiben. Read Replicas verbessern die Leseskalierbarkeit und schützen vor Datenverlust. Die Daten werden auf die Shards in einem Valkey- oder OSS Redis-Cluster (Clustermodus aktiviert) aufgeteilt.

Anwendungen verwenden den Konfigurationsendpunkt des Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert), um eine Verbindung mit den Knoten im Cluster herzustellen. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).



Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mit mehreren Shards und Replikatknoten

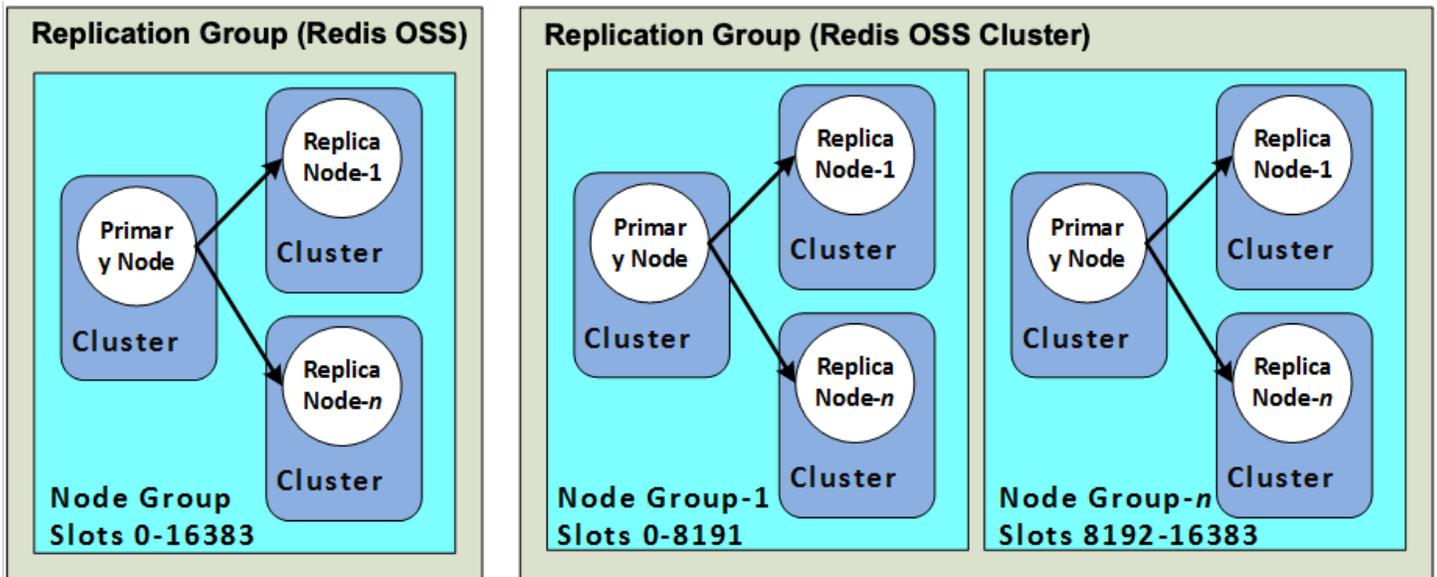
Alle Knoten in einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) müssen sich in derselben Region befinden. Zur Verbesserung der Fehlertoleranz können Sie sowohl primären Knoten als auch Read Replicas in mehreren Availability Zones innerhalb dieser Region bereitstellen.

Derzeit weisen die Funktionen von Valkey oder Redis OSS (Clustermodus aktiviert) einige Einschränkungen auf.

- Keine der Replikatknoten können zum primären Knoten manuell heraufgestuft werden.

Replikation: Valkey- und OSS Redis-Clustermodus deaktiviert oder aktiviert

Ab Valkey 7.2 und Redis OSS Version 3.2 haben Sie die Möglichkeit, einen von zwei unterschiedlichen Clustertypen (API/CLI: Replikationsgruppen) zu erstellen. Ein Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) hat immer einen einzelnen Shard (API/CLI: Knotengruppe) mit bis zu 5 Read Replica-Knoten. Ein Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) hat bis zu 500 Shards mit jeweils 1 bis 5 Read Replica-Knoten.



Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) und Valkey- oder Redis-Cluster (Clustermodus aktiviert) OSS

In der folgenden Tabelle werden wichtige Unterschiede zwischen Valkey- oder Redis-Clustern OSS (Clustermodus deaktiviert) und Valkey- oder Redis-Clustern (Clustermodus aktiviert) zusammengefasst. OSS

Vergleich von Valkey- oder Redis-Clustern OSS (Cluster-Modus deaktiviert) und Valkey- oder Redis-Clustern (Cluster-Modus aktiviert) OSS

| Funktion             | Valkey oder OSS Redis (Clustermodus deaktiviert)                                                    | Valkey oder Redis OSS (Clustermodus aktiviert)                                                                                                                                                        |
|----------------------|-----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Anpassbar            | Ja. Unterstützt das Hinzufügen und Löschen von Replikatknoten und das Hochskalieren des Knotentyps. | Begrenzt. Weitere Informationen erhalten Sie unter <a href="#">Versionsverwaltung für ElastiCache</a> und <a href="#">Skalierung von Clustern in Valkey oder Redis OSS (Clustermodus aktiviert)</a> . |
| Datenpartitionierung | Nein                                                                                                | Ja                                                                                                                                                                                                    |
| Shards               | 1                                                                                                   | 1 bis 500                                                                                                                                                                                             |
| Read Replicas        | 0 bis 5                                                                                             | 0 bis 5 pro Shard.                                                                                                                                                                                    |

| Funktion              | Valkey oder OSS Redis<br>(Clustermodus deaktiviert)                                                                                | Valkey oder Redis OSS<br>(Clustermodus aktiviert)                                                                                                                     |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <p><b>⚠ Important</b></p> <p>Wenn Sie bei einem Knotenausfall über keine Replikate verfügen, gehen alle Ihre Daten verloren.</p>   | <p><b>⚠ Important</b></p> <p>Wenn Sie bei einem Knotenausfall über keine Replikate verfügen, gehen alle Ihre Daten in diesem Shard verloren.</p>                      |
| Multi-AZ              | <p>Ja, mit mindestens 1 Replikat.</p> <p>Optional. Standardmäßig aktiviert</p>                                                     | <p>Ja</p> <p>Optional. Standardmäßig aktiviert</p>                                                                                                                    |
| Snapshots (Backups)   | <p>Ja, es wird eine einzelne .rdb-Datei erstellt.</p>                                                                              | <p>Ja, es wird für jeden Shard eine eindeutige .rdb-Datei erstellt.</p>                                                                                               |
| Wiederherstellung     | <p>Ja, mit einer einzigen .rdb-Datei aus einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert).</p>                      | <p>Ja, mit .rdb-Dateien aus einem Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) oder einem Valkey- oder Redis-Cluster (Cluster-Modus aktiviert). OSS</p> |
| Unterstützt von       | <p>Alle OSS Valkey- und Redis-Versionen</p>                                                                                        | <p>Alle Valkey-Versionen und Redis 3.2 und folgende OSS</p>                                                                                                           |
| Engine aktualisierbar | <p>Ja, mit einigen Einschränkungen. Weitere Informationen finden Sie unter <a href="#">Versionsverwaltung für ElastiCache</a>.</p> | <p>Ja, mit einigen Einschränkungen. Weitere Informationen finden Sie unter <a href="#">Versionsverwaltung für ElastiCache</a>.</p>                                    |

| Funktion                  | Valkey oder OSS Redis (Clustermodus deaktiviert)                                                                                               | Valkey oder Redis OSS (Clustermodus aktiviert)                                                                                                |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Verschlüsselung           | Versionen 3.2.6 (geplant EOL, siehe <a href="#">Zeitplan für das Ende des Lebenszyklus der OSS Redis-Versionen</a> ) und 4.0.10 und höher.     | Versionen 3.2.6 (geplant für EOL, siehe <a href="#">Zeitplan für das Ende der Lebensdauer der OSS Redis-Versionen</a> ) und 4.0.10 und höher. |
| HIPAA Teilnahmeberechtigt | Versionen 3.2.6 (geplant für EOL, siehe <a href="#">Zeitplan für das Ende des Lebenszyklus der OSS Redis-Versionen</a> ) und 4.0.10 und höher. | Versionen 3.2.6 (geplant für EOL, siehe <a href="#">Zeitplan für das Ende der Lebensdauer der OSS Redis-Versionen</a> ) und 4.0.10 und höher. |
| PCIDSS Konform            | Versionen 3.2.6 (geplant für EOL, siehe <a href="#">Zeitplan für das Ende des Lebenszyklus der OSS Redis-Versionen</a> ) und 4.0.10 und höher. | Versionen 3.2.6 (geplant für EOL, siehe <a href="#">Zeitplan für das Ende der Lebensdauer der OSS Redis-Versionen</a> ) und 4.0.10 und höher. |
| Online-Resharding         | N/A                                                                                                                                            | Version 3.2.10 (geplant für EOL, siehe <a href="#">Zeitplan für das Ende der Lebensdauer der OSS Redis-Versionen</a> ) und höher.             |

## Welchen Cluster soll ich wählen?

Wenn Sie zwischen Valkey oder Redis OSS (Cluster-Modus deaktiviert) oder Valkey oder Redis OSS (Cluster-Modus aktiviert) wählen, sollten Sie die folgenden Faktoren berücksichtigen:

- Skalierung im Vergleich zu Partitionierung – Geschäftliche Anforderungen ändern sich. Sie müssen entweder für Spitzenbedarf vorsorgen oder bei Bedarfsänderungen skalieren. Valkey oder Redis OSS (Clustermodus deaktiviert) unterstützen die Skalierung. Sie können die Lesekapazität durch Hinzufügen oder Löschen von Replikatknoten skalieren. Sie können die Kapazität auch nach oben skalieren, indem Sie auf einen größeren Knotentyp skalieren. Beide dieser Operationen sind

zeitaufwendig. Weitere Informationen finden Sie unter [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

Valkey oder Redis OSS (Clustermodus aktiviert) unterstützen die Partitionierung Ihrer Daten auf bis zu 500 Knotengruppen. Sie können die Anzahl der Shards dynamisch ändern, wenn sich Ihre Geschäftsanforderungen ändern. Ein Vorteil der Partitionierung liegt darin, dass Ihre Arbeitslast über eine größere Anzahl von Endpunkten verteilt wird. Dadurch lassen sich Engpässe in Zeiten mit höchster Auslastung reduzieren. Zusätzlich dazu können Sie einen größeren Datensatz bewältigen, da die Daten über mehrere Server verteilt werden können. Informationen zur Skalierung Ihrer Partitionen finden Sie unter [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#).

- Knotengröße vs. Anzahl der Knoten — Da ein Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) nur einen Shard hat, muss der Knotentyp groß genug sein, um alle Daten des Clusters zuzüglich des erforderlichen Overheads aufzunehmen. Da Sie Ihre Daten andererseits auf mehrere Shards partitionieren können, wenn Sie einen Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert) verwenden, können die Knotentypen kleiner sein, obwohl Sie mehr davon benötigen. Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).
- Lese- und Schreibvorgänge — Wenn die Hauptlast auf Ihrem Cluster darin besteht, dass Anwendungen Daten lesen, können Sie einen Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) skalieren, indem Sie Read Replicas hinzufügen und löschen. Es gibt jedoch maximal 5 Read Replicas. Wenn die Last auf Ihrem Cluster viele Schreibvorgänge erfordert, können Sie von den zusätzlichen Schreibendpunkten eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) mit mehreren Shards profitieren.

Unabhängig von dem ausgewählten, zu implementierenden Clustertyp ist es wichtig, einen Knotentyp zu wählen, der für Ihre derzeitigen und zukünftigen Anforderungen angemessen ist.

## Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS

Es gibt eine Reihe von Fällen, in denen ElastiCache bei Valkey und Redis OSS möglicherweise ein primärer Knoten ausgetauscht werden muss. Dazu gehören bestimmte Arten von geplanten Wartungsarbeiten und der unwahrscheinliche Fall eines Ausfalls eines Primärknotens oder einer Availability Zone.

Dieser Austausch führt zu einer gewissen Ausfallzeit für den Cluster, aber wenn Multi-AZ aktiviert ist, wird die Ausfallzeit minimiert. Die Rolle des primären Knotens wird automatisch auf eines der Read Replicas übertragen. Es ist nicht erforderlich, einen neuen Primärknoten zu erstellen und bereitzustellen, da dies transparent ElastiCache gehandhabt wird. Dieser Failover und die Replikatheraufstufung stellen sicher, dass Sie weiter in den neuen primären Knoten schreiben können, sobald die Heraufstufung abgeschlossen wurde.

ElastiCache gibt auch den Namen des Domain Name Service (DNS) des beworbenen Replikats weiter. Auf diese Weise ist in Ihrer Anwendung, falls sie in den primären Endpunkt schreibt, keine Endpunktänderung erforderlich. Wenn Sie aus individuellen Endpunkten lesen, müssen Sie den Leseendpunkt des zum primären Knoten heraufgestuften Replikats in den Endpunkt des neuen Replikats ändern.

Im Falle eines geplanten Knotenaustauschs, der aufgrund von Wartungs- oder Self-Service-Aktualisierungen eingeleitet wird, beachten Sie Folgendes:

- Bei ElastiCache Valkey- und OSS Redis-Clustern werden die geplanten Knotenersetzungen abgeschlossen, während der Cluster eingehende Schreibenanforderungen bearbeitet.
- Bei deaktivierten OSS Clustern im Valkey- und Redis-Clustermodus mit aktiviertem Multi-AZ, die auf der Engine 5.0.6 oder höher ausgeführt werden, werden die geplanten Knotenersetzungen abgeschlossen, während der Cluster eingehende Schreibenanforderungen bearbeitet.
- Bei deaktivierten OSS Clustern im Valkey- und Redis-Clustermodus mit aktiviertem Multi-AZ, die auf der Engine 4.0.10 oder früher ausgeführt werden, stellen Sie möglicherweise eine kurze Schreibunterbrechung im Zusammenhang mit Updates fest. DNS Diese Unterbrechung kann bis zu einigen Sekunden dauern. Dieser Vorgang ist viel schneller als die Neuerstellung und Bereitstellung einer neuen Primärdatenbank, was geschieht, wenn Sie Multi-AZ nicht aktivieren.

Sie können Multi-AZ mithilfe der ElastiCache Management Console, dem oder dem aktivieren. AWS CLI ElastiCache API

Die Aktivierung von ElastiCache Multi-AZ auf Ihrem Valkey- oder OSS Redis-Cluster (in der Replikationsgruppe API und CLI) verbessert Ihre Fehlertoleranz. Dies gilt insbesondere dann, wenn der Ihrem Cluster zugehörige primäre Cluster für Lese- und Schreibvorgänge nicht verfügbar ist oder ausfällt. Multi-AZ wird nur auf Valkey- und OSS Redis-Clustern mit mehr als einem Knoten in jedem Shard unterstützt.

## Themen

- [Aktivieren von Multi-AZ](#)
- [Fehlerszenarien mit Multi-AZ-Antworten](#)
- [Testen des automatischen Failovers](#)
- [Einschränkungen bei Multi-AZ](#)

## Aktivieren von Multi-AZ

Sie können Multi-AZ aktivieren, wenn Sie einen Cluster (oder eine Replikationsgruppe) mithilfe der ElastiCache Konsole API oder CLI erstellen oder ändern. AWS CLI ElastiCache API

Sie können Multi-AZ nur auf Valkey- oder Redis-Clustern OSS (Clustermodus deaktiviert) aktivieren, für die mindestens eine Read Replica verfügbar ist. Cluster ohne Read Replicas bieten keine hohe Verfügbarkeit oder Fehlertoleranz. Weitere Informationen zum Erstellen eines Clusters mit Replikation finden Sie unter [Eine Valkey- oder OSS Redis-Replikationsgruppe erstellen](#). Weitere Informationen zum Hinzufügen einer Read Replica zu einem Cluster mit Replikation finden Sie unter [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

## Themen

- [Aktivieren von Multi-AZ \(Konsole\)](#)
- [Aktivieren von Multi-AZ \(AWS CLI\)](#)
- [Aktivieren von Multi-AZ \(ElastiCache API\)](#)

## Aktivieren von Multi-AZ (Konsole)

Sie können Multi-AZ mithilfe der ElastiCache Konsole aktivieren, wenn Sie einen neuen Valkey- oder OSS Redis-Cluster erstellen oder indem Sie einen vorhandenen Cluster mit Replikation ändern.

Multi-AZ ist standardmäßig auf Valkey- oder Redis-Clustern OSS (Clustermodus aktiviert) aktiviert.

**⚠ Important**

ElastiCache aktiviert Multi-AZ automatisch nur dann, wenn der Cluster in allen Shards mindestens ein Replikat in einer anderen Availability Zone als der primären enthält.

Multi-AZ wird aktiviert, wenn ein Cluster mithilfe der Konsole erstellt wird ElastiCache

Weitere Informationen zu diesem Vorgang finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#). Achten Sie darauf, dass mehr als ein Replikat vorhanden und Multi-AZ aktiviert ist.

Aktivieren von Multi-AZ auf einem vorhandenen Cluster (Konsole)

Weitere Informationen zu diesem Prozess finden Sie unter „Ändern eines Clusters“ [Unter Verwendung der ElastiCache AWS Management Console](#).

Aktivieren von Multi-AZ (AWS CLI)

Das folgende Codebeispiel verwendet den AWS CLI , um Multi-AZ für die Replikationsgruppe zu aktivieren. `redis12`

**⚠ Important**

Die Replikationsgruppe `redis12` muss bereits vorhanden sein und mindestens eine verfügbare Read Replica besitzen.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id redis12 \
 --automatic-failover-enabled \
 --multi-az-enabled \
 --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id redis12 ^
```

```
--automatic-failover-enabled ^
--multi-az-enabled ^
--apply-immediately
```

Die JSON Ausgabe dieses Befehls sollte etwa wie folgt aussehen.

```
{
 "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": {}
}
}
```

Weitere Informationen zu diesen Themen finden Sie in der AWS CLI -Befehlsreferenz:

- [create-cache-cluster](#)
- [create-replication-group](#)
- [modify-replication-group](#) in der AWS CLI Befehlsreferenz.

### Aktivieren von Multi-AZ (ElastiCache API)

Das folgende Codebeispiel verwendet den ElastiCache API, um Multi-AZ für die Replikationsgruppe `redis12` zu aktivieren.

#### Note

Zur Verwendung dieses Beispiels muss die Replikationsgruppe `redis12` bereits vorhanden sein und mindestens ein verfügbares Lesereplikat besitzen.

```
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>
```

Weitere Informationen finden Sie in der ElastiCache APIReferenz zu den folgenden Themen:

- [CreateCacheCluster](#)
- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

## Fehlerszenarien mit Multi-AZ-Antworten

Vor der Einführung von Multi-AZ wurden die ausgefallenen Knoten eines Clusters ElastiCache erkannt und ersetzt, indem der ausgefallene Knoten neu erstellt und neu bereitgestellt wurde. Wenn Sie Multi-AZ aktivieren, wird bei einem ausgefallenen primärer Knoten ein Failover auf das Replikat mit der geringsten Replikationsverzögerung durchgeführt. Das ausgewählte Replikat wird automatisch zum primären Knoten heraufgestuft. Dies ist sehr viel schneller, als einen neuen primären Knoten zu erstellen und neu bereitzustellen. Bei diesem Vorgang dauert gewöhnlich nur wenige Sekunden, bis Sie wieder in den Cluster schreiben können.

Wenn Multi-AZ aktiviert ist, wird der Status des ElastiCache Primärknotens kontinuierlich überwacht. Sollte der primäre Knoten ausfallen, wird abhängig von der Art des Ausfalls eine der folgenden Aktionen durchgeführt.

### Themen

- [Fehlerszenarien, wenn nur der Primärknoten ausfällt](#)
- [Fehlerszenarien, wenn der Primärknoten und einige Read Replicas \(Lesereplikate\) fehlschlagen](#)
- [Fehlerszenarien, wenn der gesamte Cluster ausfällt](#)

### Fehlerszenarien, wenn nur der Primärknoten ausfällt

Wenn nur der primäre Knoten ausfällt, wird das Read Replica mit der geringsten Replikationsverzögerung zum primären Cluster heraufgestuft. Ein Ersatz-Read Replica wird dann in derselben Availability Zone wie der ausgefallene primäre Knoten erstellt und bereitgestellt .

Wenn nur der Primärknoten ausfällt, geht ElastiCache Multi-AZ wie folgt vor:

1. Der ausgefallene primäre Knoten wird in den Offline-Zustand versetzt.
2. Die Read Replica mit der geringsten Replikationsverzögerung wird zum primären Knoten heraufgestuft.

Die Schreibvorgänge können fortgesetzt werden, sobald der Vorgang der Heraufstufung abgeschlossen ist. Dies dauert in der Regel nur einige Sekunden. Wenn Ihre Anwendung auf den primären Endpunkt schreibt, müssen Sie den Endpunkt für Schreib- oder Lesevorgänge nicht ändern. ElastiCache gibt den DNS Namen des hochgestuften Replikats weiter.

3. Ein Ersatz-Read Replica wird gestartet und bereitgestellt.

Die Ersatz-Read Replica wird in der Availability Zone gestartet, in der sich der ausgefallene Knoten befand. Die Verteilung der Knoten bleibt daher erhalten.

4. Die anderen Replikate werden mit dem neuen primären Knoten synchronisiert.

Nachdem das neue Replikat verfügbar ist, beachten Sie die folgenden Effekte:

- Primärer Endpunkt — Sie müssen keine Änderungen an Ihrer Anwendung vornehmen, da der DNS Name des neuen primären Knotens an den primären Endpunkt weitergegeben wird.
- Lese-Endpunkt – Der Lese-Endpunkt wird automatisch aktualisiert, sodass er auf die neuen Replikatknoten verweist.

Weitere Informationen zum Suchen der Endpunkte eines Clusters finden Sie in den folgenden Themen:

- [Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder OSS Redis-Replikationsgruppen \(\)AWS CLI](#)
- [Suchen nach Endpunkten für Valkey- oder OSS Redis-Replikationsgruppen \(\) ElastiCache API](#)

Fehlerszenarien, wenn der Primärknoten und einige Read Replicas (Lesereplikate) fehlschlagen

Bei Ausfall des primären Clusters und mindestens einer Read Replica wird das verfügbare Replikat mit der geringsten Replikationsverzögerung zum primären Cluster heraufgestuft. Zudem werden in denselben Availability Zones, in der sich die ausgefallenen Knoten und das zum primären Cluster heraufgestufte Replikat befanden, neue Read Replicas erstellt und bereitgestellt.

Wenn der Primärknoten und einige Read Replicas ausfallen, geht ElastiCache Multi-AZ wie folgt vor:

1. Der ausgefallene primäre Knoten und ausgefallenen die Read Replicas werden in den Offline-Zustand versetzt.
2. Das verfügbare Replikat mit der geringsten Replikationsverzögerung wird zum primären Knoten heraufgestuft.

Die Schreibvorgänge können fortgesetzt werden, sobald der Vorgang der Heraufstufung abgeschlossen ist. Dies dauert in der Regel nur einige Sekunden. Wenn Ihre Anwendung auf

den primären Endpunkt schreibt, müssen Sie den Endpunkt für Schreibvorgänge nicht ändern. ElastiCache propagiert den DNS Namen des hochgestuften Replikats.

### 3. Ersatzreplikate werden erstellt und bereitgestellt.

Die Ersatzreplikate werden in den Availability Zones der ausgefallenen Knoten erstellt, sodass die Verteilung der Knoten erhalten bleibt.

### 4. Alle Cluster werden mit dem neuen primären Knoten synchronisiert.

Nehmen Sie die folgenden Änderungen an Ihrer Anwendung vor, wenn die neuen Knoten verfügbar sind:

- Primärer Endpunkt – Nehmen Sie keine Änderungen an Ihrer Anwendung vor. Der DNS Name des neuen Primärknotens wird an den primären Endpunkt weitergegeben.
- Lese-Endpunkt – Der Lese-Endpunkt wird automatisch so aktualisiert, dass er auf die neuen Replikatknoten verweist.

Weitere Informationen zum Suchen der Endpunkte einer Replikationsgruppe finden Sie in den folgenden Themen:

- [Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder OSS Redis-Replikationsgruppen \(\)AWS CLI](#)
- [Suchen nach Endpunkten für Valkey- oder OSS Redis-Replikationsgruppen \(\) ElastiCache API](#)

## Fehlerszenarien, wenn der gesamte Cluster ausfällt

Bei einem umfassenden Ausfall werden in denselben Availability Zones, der sich die Originalknoten befanden, alle Knoten neu erstellt und bereitgestellt.

In diesem Szenario gehen alle Daten im Cluster aufgrund des Ausfalls eines jeden Knotens im Cluster verloren. Ein solches Ereignis ist selten.

Wenn der gesamte Cluster ausfällt, geht ElastiCache Multi-AZ wie folgt vor:

1. Der ausgefallene primäre Knoten und die Read Replicas werden in den Offline-Zustand versetzt.

2. Es wird ein primäre Ersatzknoten erstellt und bereitgestellt.
3. Ersatzreplikate werden erstellt und bereitgestellt.

Die Ersetzungen werden in den Availability Zones der ausgefallenen Knoten erstellt, sodass die Verteilung der Knoten erhalten bleibt.

Da der gesamte Cluster ausgefallen ist, kam es zu Datenverlust. Alle neuen Knoten werden kalt gestartet.

Da jeder Ersatzknoten denselben Endpunkt wie der Knoten hat, der durch ihn ersetzt wird, müssen in Ihrer Anwendung keine Endpunktänderungen vorgenommen werden.

Weitere Informationen zum Suchen der Endpunkte einer Replikationsgruppe finden Sie in den folgenden Themen:

- [Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder OSS Redis-Replikationsgruppen \(\)AWS CLI](#)
- [Suchen nach Endpunkten für Valkey- oder OSS Redis-Replikationsgruppen \(\) ElastiCache API](#)

Es wird empfohlen, den primären Knoten und die Read Replicas in verschiedenen Availability Zones zu erstellen. Dadurch wird der Grad der Fehlertoleranz erhöht.

## Testen des automatischen Failovers

Nachdem Sie das automatische Failover aktiviert haben, können Sie es mit der ElastiCache Konsole, der AWS CLI, dem `test-failover`-Befehl oder der ElastiCache API testen.

Beim Testen ist Folgendes zu beachten:

- Sie können diesen Vorgang verwenden, um das automatische Failover auf bis zu 15 Shards (im ElastiCache API und als Knotengruppen bezeichnet AWS CLI) in einem beliebigen Zeitraum von 24 Stunden zu testen.
- Wenn Sie diesen Vorgang für Shards in verschiedenen Clustern (im API und als Replikationsgruppen bezeichnet CLI) aufrufen, können Sie die Aufrufe gleichzeitig tätigen.
- In einigen Fällen können Sie diesen Vorgang mehrmals auf verschiedenen Shards in derselben Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) aufrufen. In solchen Fällen muss die erste Knotenersetzung abgeschlossen werden, bevor ein nachfolgender Aufruf ausgeführt werden kann.
- Um festzustellen, ob der Knotenaustausch abgeschlossen ist, überprüfen Sie die Ereignisse mithilfe der ElastiCache Amazon-Konsole, der AWS CLI, der oder der ElastiCache API. Suchen Sie nach den folgenden Ereignissen im Zusammenhang mit automatischem Failover, die hier nach der Wahrscheinlichkeit ihres Auftretens aufgelistet werden:
  1. Meldung der Replikationsgruppe: `Test Failover API called for node group <node-group-id>`
  2. Meldung des Cache-Clusters: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  3. Meldung der Replikationsgruppe: `Failover from primary node <primary-node-id> to replica node <node-id> completed`
  4. Meldung des Cache-Clusters: `Recovering cache nodes <node-id>`
  5. Meldung des Cache-Clusters: `Finished recovery for cache nodes <node-id>`

Weitere Informationen finden Sie hier:

- [ElastiCache Ereignisse anzeigen](#) im ElastiCache -Benutzerhandbuch
- [DescribeEvents](#) in der ElastiCache API Referenz
- [describe-events](#) in der AWS CLI -Befehlsreferenz.
- Dies API dient dazu, das Verhalten Ihrer Anwendung im Falle eines ElastiCache Failovers zu testen. Sie wurde nicht als Betriebssystem zum Einleiten eines Failovers konzipiert, um ein Problem

mit dem Cluster zu beheben. Darüber hinaus AWS kann dies unter bestimmten Bedingungen, z. B. bei großen Betriebsereignissen, blockiert werden. API

## Themen

- [Testen des automatischen Failovers mit dem AWS Management Console](#)
- [Testen Sie den automatischen Failover mit dem AWS CLI](#)
- [Testen Sie den automatischen Failover mit dem ElastiCache API](#)

## Testen des automatischen Failovers mit dem AWS Management Console

Verwenden Sie das folgende Verfahren, um das automatische Failover mit der Konsole zu testen.

So testen Sie das automatische Failover

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Valkey oder OSSRedis aus.
3. Wählen Sie in der Clusterliste das Feld links neben dem Cluster aus, den Sie testen möchten. Dieser Cluster muss mindestens einen Read Replica-Knoten enthalten.
4. Bestätigen Sie im Bereich Details, dass dieser Cluster Multi-AZ-fähig ist. Wenn der Cluster nicht Multi-AZ-fähig ist, wählen Sie einen anderen Cluster aus oder bearbeiten Sie diesen Cluster so, dass Multi-AZ aktiviert wird. Weitere Informationen finden Sie unter [Unter Verwendung der ElastiCache AWS Management Console](#).



5. Wählen Sie für Valkey oder Redis OSS (Clustermodus deaktiviert) den Namen des Clusters.

Gehen Sie für Valkey oder Redis OSS (Clustermodus aktiviert) wie folgt vor:

- a. Wählen Sie den Cluster-Namen aus.
  - b. Wählen Sie auf der Seite Shards für den Shard (im API und als Knotengruppe bezeichnet CLI), auf dem Sie den Failover testen möchten, den Namen des Shards aus.
6. Wählen Sie auf der Seite „Nodes“ Failover Primary.
  7. Wählen Sie Continue, um ein Failover des primären Knotens auszuführen, oder wählen Sie Cancel, um die Operation ohne ein Failover des primären Knotens abzubrechen.

Während des Failover-Vorgangs zeigt die Konsole den Status des Knotens weiterhin als available an. Um den Status des Failover-Tests zu verfolgen, wählen Sie im Navigationsbereich der Konsole Events aus. Suchen Sie auf der Registerkarte Events nach Ereignissen, für die angegeben wird, dass Ihr Failover gestartet (Test Failover API called) und abgeschlossen (Recovery completed) wurde.

Testen Sie den automatischen Failover mit dem AWS CLI

Mithilfe dieses Vorgangs können Sie das automatische Failover auf jedem Multi-AZ-fähigen Cluster testen. `AWS CLI test-failover`

Parameter

- `--replication-group-id` – Erforderlich. Die Replikationsgruppe (auf der Konsole als Cluster bezeichnet), die getestet werden soll.
- `--node-group-id` – Erforderlich. Der Name der Knotengruppe, auf der das automatische Failover getestet werden soll. Sie können maximal 15 Knotengruppen in einem fortlaufenden Zeitraum von 24 Stunden testen.

Im folgenden Beispiel wird der verwendet AWS CLI , um den automatischen Failover für die Knotengruppe `redis00-0003` im Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) zu testen. `redis00`

Example Testen des automatischen Failovers

Für Linux, macOS oder Unix:

```
aws elasticache test-failover \
 --replication-group-id redis00 \
 --node-group-id redis00-0003
```

Für Windows:

```
aws elasticache test-failover ^
 --replication-group-id redis00 ^
 --node-group-id redis00-0003
```

Die Ausgabe des vorhergehenden Befehls sieht in etwa wie folgt aus.

```
{
 "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": {}
}
}

```

Verwenden Sie den Vorgang, um den Fortschritt Ihres Failovers zu verfolgen. AWS CLI `describe-events`

Weitere Informationen finden Sie hier:

- [test-failover](#) in der AWS CLI -Befehlsreferenz.

- [describe-events](#) in der AWS CLI -Befehlsreferenz.

Testen Sie den automatischen Failover mit dem ElastiCache API

Mithilfe dieses Vorgangs können Sie das automatische Failover auf jedem Cluster testen, für den ElastiCache API Multi-AZ aktiviert ist. `TestFailover`

#### Parameter

- `ReplicationGroupId` – Erforderlich. Die Replikationsgruppe (auf der Konsole als Cluster bezeichnet), die getestet werden soll.
- `NodeGroupId` – Erforderlich. Der Name der Knotengruppe, auf der das automatische Failover getestet werden soll. Sie können maximal 15 Knotengruppen in einem fortlaufenden Zeitraum von 24 Stunden testen.

Das folgende Beispiel testet das automatische Failover auf der Knotengruppe `redis00-0003` in der Replikationsgruppe (auf der Konsole im Cluster) `redis00`.

#### Example Testen des automatischen Failovers

```
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>
```

Verwenden Sie den ElastiCache `DescribeEvents` API Vorgang, um den Fortschritt Ihres Failovers zu verfolgen.

Weitere Informationen finden Sie hier:

- [TestFailover](#) in der Referenz ElastiCache API
- [DescribeEvents](#) in der ElastiCache APIReferenz

## Einschränkungen bei Multi-AZ

Beachten Sie die folgenden Einschränkungen für Multi-AZ:

- Multi-AZ wird auf Valkey und auf Redis OSS Version 2.8.6 und höher unterstützt.
- Multi-AZ wird auf T1-Knotentypen nicht unterstützt.
- Die Valkey- und OSS Redis-Replikation erfolgt asynchron. Daher kann es passieren, dass bei dem Failover eines primären Clusters auf ein Replikat aufgrund der Replikationsverzögerung eine kleine Datenmenge verloren geht.

Bei der Auswahl des Replikats, das zum Primärreplikat heraufgestuft werden soll, wird das Replikat mit ElastiCache der geringsten Verzögerung bei der Replikation ausgewählt. Es wird also das neueste Replikat ausgewählt. Hierdurch kann die Menge verlorener Daten reduziert werden. Das Replikat mit der geringsten Replikationsverzögerung kann sich in der gleichen oder in einer anderen Availability Zone als der ausgefallene primäre Knoten befinden.

- Wenn Sie Read Replicas auf Valkey- oder OSS Redis-Clustern mit deaktiviertem Clustermodus manuell zu primär heraufstufen, können Sie dies nur tun, wenn Multi-AZ und automatisches Failover deaktiviert sind. So stufen Sie eine Read Replica zum primären Knoten herauf:
  1. Deaktivieren Sie Multi-AZ auf dem Cluster.
  2. Deaktivieren Sie das automatische Failover auf dem Cluster. Sie können dies über die Konsole tun, indem Sie das Kontrollkästchen Auto Failover für die Replikationsgruppe deaktivieren. Sie können dies auch tun, AWS CLI indem Sie die `AutomaticFailoverEnabled` Eigenschaft `false` beim Aufrufen des `ModifyReplicationGroup` Vorgangs auf setzen.
  3. Stufen Sie das Lesereplikat zum primären Knoten herauf.
  4. Multi-AZ wieder aktivieren.
- ElastiCache (RedisOSS) Multi-AZ und Datei (AOF), die nur anhängen, schließen sich gegenseitig aus. Wenn Sie eine dieser Funktionen aktivieren, kann die andere nicht aktiviert werden.
- Der Ausfall eines Knotens kann durch den Ausfall einer gesamten Availability Zone verursacht werden. In diesem Fall wird das Replikat, das den ausgefallenen primären Knoten ersetzt, nur bei Verfügbarkeit der Availability Zone erstellt. Betrachten Sie beispielsweise eine Replikationsgruppe mit den primären Knoten in AZ-a und Replikaten in AZ-b und AZ-c. Wenn der primäre Knoten ausfällt, wird das Replikat mit der geringsten Replikationsverzögerung zum primären Cluster heraufgestuft. ElastiCache Erstellt dann nur dann ein neues Replikat in AZ-a (wo sich die ausgefallene Primärreplikation befand), wenn AZ-a gesichert und verfügbar ist.

- Ein vom Kunden initiiertes Neustart des primären Clusters löst kein automatisches Failover aus. Andere Neustarts und Ausfälle hingegen lösen ein automatisches Failover aus.
- Wenn der primäre Cluster neu gestartet wird, werden seine Daten gelöscht, wenn er wieder online ist. Wenn die Read Replicas den gelöschten primären Cluster sehen, löschen sie ihre Kopie der Daten, was zu Datenverlust führt.
- Nachdem die Read Replica heraufgestuft wurde, werden die anderen Replikate mit dem neuen primären Cluster synchronisiert. Nach der ersten Synchronisierung wird der Inhalt der Replikate gelöscht, und sie synchronisieren die Daten aus dem neuen primären Cluster. Dieser Synchronisierungsvorgang führt zu einer kurzen Unterbrechung, bei der auf die Replikate nicht zugegriffen werden kann. Dieser Synchronisierungsvorgang bewirkt eine temporäre Zunahme der Arbeitslast auf dem primären Cluster, während er mit den Replikaten synchronisiert wird. Dieses Verhalten ist typisch für Valkey und Redis OSS und gilt nicht nur für Multi-AZ. ElastiCache Einzelheiten zu diesem Verhalten finden Sie unter [Replikation](#) auf der Valkey-Website.

#### Wichtig

Für Valkey 7.2.6 und höher oder Redis OSS Version 2.8.22 und höher können Sie keine externen Replikate erstellen.

Für OSS Redis-Versionen vor 2.8.22 empfehlen wir, kein externes Replikat mit einem Cluster zu verbinden, der Multi-AZ-fähig ist. ElastiCache Diese nicht unterstützte Konfiguration kann zu Problemen führen, die eine ordnungsgemäße Durchführung von Failover und Wiederherstellung ElastiCache verhindern. Um ein externes Replikat mit einem ElastiCache Cluster zu verbinden, stellen Sie sicher, dass Multi-AZ nicht aktiviert ist, bevor Sie die Verbindung herstellen.

## So werden Synchronisation und Backup implementiert

Alle unterstützten Versionen von Valkey und Redis OSS unterstützen Backup und Synchronisation zwischen dem Primär- und dem Replikatknoten. Die Art und Weise, wie Sicherung und Synchronisation implementiert werden, variiert jedoch je nach Version.

### Redis OSS Version 2.8.22 und höher

Bei der OSS Redis-Replikation in den Versionen 2.8.22 und höher können Sie zwischen zwei Methoden wählen. Weitere Informationen erhalten Sie unter [Redis-Versionen OSS vor 2.8.22](#) und [Snapshot und Wiederherstellung](#).

Wenn die Schreiblasten während des unvergabelten Vorgangs zu intensiv werden, werden Schreibvorgänge in den Cluster verzögert, damit sich nicht zu viele Änderungen anstauen und somit einen erfolgreichen Snapshot verhindern würden.

### Redis-Versionen OSS vor 2.8.22

Die OSS Sicherung und Synchronisation von Redis in Versionen vor 2.8.22 erfolgt in drei Schritten.

1. Vergabeln und serialisieren Sie die Clusterdaten als Hintergrundprozess auf dem Datenträger. Dadurch wird ein Snapshot erstellt. point-in-time
2. Akkumulieren Sie im Vordergrund ein Änderungsprotokoll im Client-Ausgabepuffer.

#### Important

Wenn das Änderungsprotokoll die Größe des Client-Ausgabepuffers überschreitet, schlägt die Sicherung oder Synchronisierung fehl. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#).

3. Übertragen Sie schließlich die Cachedaten und anschließend das Änderungsprotokoll auf den Replikatknoten.

## Eine Valkey- oder OSS Redis-Replikationsgruppe erstellen

Es bieten sich Ihnen die folgenden Optionen für das Erstellen eines Clusters mit Replikatknoten. Eine Option gilt, wenn Sie bereits über einen verfügbaren Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) verfügen, der keinem Cluster zugeordnet ist, der über Replikate verfügt, die als primärer Knoten verwendet werden können. Die andere trifft zu, wenn Sie einen Primärknoten mit dem Cluster und Lesereplikaten erstellen müssen. Derzeit muss ein Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) von Grund auf neu erstellt werden.

### Option 1: [Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters](#)

Verwenden Sie diese Option, um einen vorhandenen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem Knoten zu nutzen. Sie geben diesen vorhandenen Knoten als Primärknoten im neuen Cluster an und fügen dem Cluster dann einzeln 1 bis 5 Lesereplikate hinzu. Wenn der vorhandene Cluster aktiv ist, werden die Read Replicas bei ihrer Erstellung mit ihm synchronisiert. Siehe [Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters](#).

#### Important

Sie können keinen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mithilfe eines vorhandenen Clusters erstellen. Informationen zum Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole finden Sie unter [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#)

### Option 2: [Eine Valkey- oder OSS Redis-Replikationsgruppe von Grund auf neu erstellen](#)

Verwenden Sie diese Option, wenn Sie noch keinen verfügbaren Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) haben, den Sie als primären Knoten des Clusters verwenden können, oder wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) erstellen möchten. Siehe [Eine Valkey- oder OSS Redis-Replikationsgruppe von Grund auf neu erstellen](#).

## Erstellen einer Replikationsgruppe unter Verwendung eines vorhandenen Clusters

Ein verfügbarer Cluster ist ein vorhandener Valkey- oder OSS Redis-Cluster mit einem Knoten. Derzeit unterstützen Valkey oder Redis OSS (Clustermodus aktiviert) die Erstellung eines Clusters mit Replikaten unter Verwendung eines verfügbaren Clusters mit einem einzelnen Knoten nicht. Informationen zum Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) finden Sie unter. [Erstellen eines Valkey- oder Redis-Clusters OSS \(Clustermodus aktiviert\) \(Konsole\)](#)

Das folgende Verfahren kann nur verwendet werden, wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem Knoten haben. Der Knoten dieses Clusters wird zum primären Knoten im neuen Cluster. Wenn Sie keinen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) haben, den Sie als primären Cluster verwenden können, finden Sie weitere Informationen unter. [Eine Valkey- oder OSS Redis-Replikationsgruppe von Grund auf neu erstellen](#)

Erstellen einer Replikationsgruppe mithilfe eines vorhandenen Clusters (Konsole)

Weitere Informationen finden Sie im Thema [Mit dem ElastiCache AWS Management Console](#).

Erstellen einer Replikationsgruppe mithilfe eines verfügbaren Valkey- oder OSS Redis-Cache-Clusters (AWS CLI)

Es gibt zwei Schritte zum Erstellen einer Replikationsgruppe mit Read Replicas, wenn Sie einen verfügbaren Valkey- oder OSS Redis-Cache-Cluster als Primär-Cache-Cluster verwenden, wenn Sie den verwenden. AWS CLI

Wenn AWS CLI Sie den verwenden, erstellen Sie eine Replikationsgruppe, indem Sie den verfügbaren Standalone-Knoten als primären Knoten des Clusters `--primary-cluster-id` und die Anzahl der Knoten, die Sie im Cluster haben möchten, mit dem CLI Befehl angeben. `create-replication-group` Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe. Der Wert dieses Parameters wird als Basis für die Namen der hinzugefügten Knoten mit einer fortlaufenden 3-stelligen Nummer am Ende der `--replication-group-id` verwendet. Beispiel, `sample-repl-group-001`.

Die Benennungsbeschränkungen für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.

- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

#### --replication-group-description

Beschreibung der Replikationsgruppe.

#### --num-node-groups

Die Anzahl der Knoten, die Sie in diesem Cluster benötigen. Dieser Wert umfasst den primären Knoten. Dieser Parameter hat einen Maximalwert von sechs.

#### --primary-cluster-id

Der Name des Knotens des verfügbaren Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert), den Sie als primärer Knoten in dieser Replikationsgruppe verwenden möchten.

Mit dem folgenden Befehl wird die Replikationsgruppe erstellt, `sample-repl-group` wobei der verfügbare Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) `redis01` als primärer Knoten der Replikationsgruppe verwendet wird. Er erstellt 2 neue Knoten, bei denen es sich um Read Replicas handelt. Die Einstellungen von `redis01` (d. h. Parametergruppe, Sicherheitsgruppe, Knotentyp, Engine-Version usw.) werden auf alle Knoten in der Replikationsgruppe angewendet.

Für Linux, macOS oder 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
```

Für 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
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im Thema AWS CLI [create-replication-group](#).

## Als Nächstes Hinzufügen von Read Replicas zur Replikationsgruppe

Nachdem die Replikationsgruppe erstellt wurde, fügen Sie ihr mit dem Befehl `create-cache-cluster` eine bis fünf Read Replicas hinzu und schließen Sie unbedingt die folgenden Parameter ein.

`--cache-cluster-id`

Der Name des Clusters, den Sie zur Replikationsgruppe hinzufügen.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`--replication-group-id`

Der Name der Replikationsgruppe, zu der Sie diesen Cache-Cluster hinzufügen.

Wiederholen Sie diesen Befehl für jede Read Replica, die Sie zur Replikationsgruppe hinzufügen möchten, und ändern Sie dabei nur den Wert des Parameters `--cache-cluster-id`.

### Note

Denken Sie daran, dass eine Replikationsgruppe über nicht mehr als fünf Read Replicas verfügen kann. Bei dem Versuch, einer Replikationsgruppe mit bereits fünf Read Replicas eine weitere Read Replica hinzuzufügen, schlägt die Operation fehl.

Der folgende Code fügt die Read Replica `my-replica01` zur Replikationsgruppe `sample-repl-group` hinzu. Die Einstellungen des primären Clusters – Parametergruppe, Sicherheitsgruppe, Knotentyp usw. – werden auf Knoten angewendet, sobald sie der Replikationsgruppe hinzugefügt werden.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \
 --cache-cluster-id my-replica01 \
 --replication-group-id sample-repl-group \
 --replicas-per-availability-zone 5
```

```
--replication-group-id sample-repl-group
```

Für Windows:

```
aws elasticache create-cache-cluster ^
 --cache-cluster-id my-replica01 ^
 --replication-group-id sample-repl-group
```

Die Ausgabe dieses Befehls sieht in etwa folgendermaßen aus.

```
{
 "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": {}
 }
}
```

Weitere Informationen finden Sie in den folgenden AWS CLI Themen:

- [create-replication-group](#)
- [modify-replication-group](#)

Hinzufügen von Replikaten zu einem eigenständigen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) () ElastiCache API

Wenn Sie den verwenden ElastiCache API, erstellen Sie eine Replikationsgruppe, die den verfügbaren Standalone-Knoten als primären Knoten des Clusters `PrimaryClusterId` und

die Anzahl der Knoten, die Sie im Cluster haben möchten, mit dem CLI Befehl, angeben. `CreateReplicationGroup` Schließen Sie die folgenden Parameter ein.

### ReplicationGroupId

Der Name der zu erstellenden Replikationsgruppe. Der Wert dieses Parameters wird als Basis für die Namen der hinzugefügten Knoten mit einer fortlaufenden 3-stelligen Nummer am Ende der `ReplicationGroupId` verwendet. Beispiel, `sample-repl-group-001`.

Die Benennungsbeschränkungen für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

### ReplicationGroupDescription

Beschreibung des Clusters mit Replikaten.

### NumCacheClusters

Die Anzahl der Knoten, die Sie in diesem Cluster benötigen. Dieser Wert umfasst den primären Knoten. Dieser Parameter hat einen Maximalwert von sechs.

### PrimaryClusterId

Der Name des verfügbaren Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert), den Sie als primärer Knoten in diesem Cluster verwenden möchten.

Der folgende Befehl erstellt den Cluster mit Replikaten, `sample-repl-group` wobei der verfügbare Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) `redis01` als primärer Knoten der Replikationsgruppe verwendet wird. Er erstellt 2 neue Knoten, bei denen es sich um Read Replicas handelt. Die Einstellungen von `redis01` (d. h. Parametergruppe, Sicherheitsgruppe, Knotentyp, Engine-Version usw.) werden auf alle Knoten in der Replikationsgruppe angewendet.

```
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>
```

Weitere Informationen finden Sie in den ElastiCache APL-Themen:

- [CreateReplicationGroup](#)
- [ModifyReplicationGroup](#)

Als Nächstes Hinzufügen von Read Replicas zur Replikationsgruppe

Nachdem die Replikationsgruppe erstellt wurde, fügen Sie ihr mit der Operation `CreateCacheCluster` eine bis fünf Read Replicas hinzu und schließen Sie unbedingt die folgenden Parameter ein.

`CacheClusterId`

Der Name des Clusters, den Sie zur Replikationsgruppe hinzufügen.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`ReplicationGroupId`

Der Name der Replikationsgruppe, zu der Sie diesen Cache-Cluster hinzufügen.

Wiederholen Sie diese Operation für jede Read Replica, die Sie zur Replikationsgruppe hinzufügen möchten, und ändern Sie dabei nur den Wert des Parameters `CacheClusterId`.

Der folgende Code fügt die Lesereplikate `myReplica01` der Replikationsgruppe `myReplGroup` hinzu. Die Einstellungen des primären Clusters – Parametergruppe, Sicherheitsgruppe, Knotentyp usw. – werden auf Knoten angewendet, wenn sie der Replikationsgruppe hinzugefügt werden.

```
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]
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im Thema ElastiCache API [CreateCacheCluster](#).

## Eine Valkey- oder OSS Redis-Replikationsgruppe von Grund auf neu erstellen

Im Folgenden erfahren Sie, wie Sie eine Valkey- oder OSS Redis-Replikationsgruppe erstellen, ohne einen vorhandenen Valkey- oder Redis-Cluster als primären Cluster zu verwenden. OSS Sie können eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) oder Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) von Grund auf neu erstellen, indem Sie die Konsole, die oder die verwenden. ElastiCache AWS CLI ElastiCache API

Bevor Sie fortfahren, entscheiden Sie, ob Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) oder eine Valkey- oder Redis-Replikationsgruppe (Clustermodus aktiviert) erstellen möchten. OSS Hilfreiche Informationen zur Entscheidung finden Sie unter [Replikation: Valkey- und OSS Redis-Clustermodus deaktiviert oder aktiviert](#).

### Themen

- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) von Grund auf neu erstellen](#)
- [Eine Replikationsgruppe in Valkey oder Redis OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen](#)

## Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) von Grund auf neu erstellen

Sie können eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) von Grund auf neu erstellen, indem Sie die ElastiCache Konsole, die oder die AWS CLI verwenden. ElastiCache API Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) besteht immer aus einer Knotengruppe, einem primären Cluster und bis zu fünf Read Replicas. Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) unterstützen die Partitionierung Ihrer Daten nicht.

### Note

Das Knoten- oder Shard-Limit kann auf einen Maximalwert von 500 pro Cluster erhöht werden. Informationen zum Anfordern einer Limiterhöhung finden Sie unter [AWS Service-Limits](#), wenn Sie den Instance-Typ in die Anfrage eingeben.

Gehen Sie wie folgt vor, um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) von Grund auf neu zu erstellen:

Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) von Grund auf neu erstellen ()AWS CLI

Das folgende Verfahren erstellt eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) mithilfe von AWS CLI

Wenn Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und all ihre Knoten mit einem einzigen Befehlsaufruf. AWS CLI `create-replication-group` Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe.

Die Benennungsbeschränkungen für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) lauten wie folgt:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.

- Er darf nicht mit einem Bindestrich enden.

`--replication-group-description`

Beschreibung der Replikationsgruppe.

`--num-cache-clusters`

Die Gesamtzahl von Knoten, die Sie mit dieser Replikationsgruppe erstellen möchten, primäre Knoten und Read Replicas kombiniert.

Wenn Sie Multi-AZ (`--automatic-failover-enabled`) aktivieren, muss der Wert für `--num-cache-clusters` mindestens 2 sein.

`--cache-node-type`

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2Amazon-Instanztypen](#).

`--data-tiering-enabled`

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

`--cache-parameter-group`

Geben Sie eine Parametergruppe an, die Ihrer Engine-Version entspricht. Wenn Sie Redis OSS 3.2.4 oder höher ausführen, geben Sie die Parametergruppe oder eine davon abgeleitete `default.redis3.2` Parametergruppe an, um eine Valkey- oder OSS Redis-Replikationsgruppe (Clustermodus deaktiviert) `default.redis3.2` zu erstellen. Weitere Informationen finden Sie unter [Valkey- und Redis-Parameter OSS](#).

`--Netzwerktyp`

Entweder `ipv4`, `ipv6` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

`--engine`

`redis`

## --engine-version

Wählen Sie für den umfangreichsten Satz von Funktionen die neueste Engine-Version aus.

Die Namen der Knoten werden vom Namen der Replikationsgruppe abgeleitet, indem `-00#` an den Namen der Replikationsgruppe angehängt wird. Wenn Sie z. B. den Replikationsgruppennamen `myReplGroup` verwenden, lauten die Namen des primären Knotens `myReplGroup-001` und der Read Replicas `myReplGroup-002` bis `myReplGroup-006`.

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `--transit-encryption-enabled` oder `--at-rest-encryption-enabled` hinzu und erfüllen die folgenden Bedingungen.

- Auf Ihrer Replikationsgruppe muss Redis Version 3.2.6 oder 4.0.10 OSS ausgeführt werden.
- Die Replikationsgruppe muss in einem Amazon erstellt werden VPC.
- Sie müssen außerdem den Parameter `--cache-subnet-group` einbinden.
- Sie müssen auch den Parameter `--auth-token` mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH Token (Passwort) angeben, das für die Ausführung von Vorgängen an dieser Replikationsgruppe erforderlich ist.

Der folgende Vorgang erstellt eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) `sample-repl-group` mit drei Knoten, einem primären und zwei Replikaten.

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe dieses Befehls sieht in etwa wie folgt aus:

```
{
 "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": {}
 }
}
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im Thema. AWS CLI [create-replication-group](#)

Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) von Grund auf neu erstellen () ElastiCache API

Das folgende Verfahren erstellt eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) mithilfe von. ElastiCache API

Wenn Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und all ihre Knoten mit einem einzigen Aufruf des Vorgangs. ElastiCache API CreateReplicationGroup Schließen Sie die folgenden Parameter ein.

ReplicationGroupId

Der Name der zu erstellenden Replikationsgruppe.

Für die Benennung von Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus aktiviert) gelten folgende Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

### ReplicationGroupDescription

Ihre Beschreibung der Replikationsgruppe.

### NumCacheClusters

Die Gesamtzahl der Knoten, die Sie mit dieser Replikationsgruppe erstellen möchten, primäre Knoten und Read Replicas kombiniert.

Wenn Sie Multi-AZ (`AutomaticFailoverEnabled=true`) aktivieren, muss der Wert für `NumCacheClusters` mindestens 2 sein.

### CacheNodeType

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2Amazon-Instanztypen](#).

### --data-tiering-enabled

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

### CacheParameterGroup

Geben Sie eine Parametergruppe an, die Ihrer Engine-Version entspricht. Wenn Sie Redis OSS 3.2.4 oder höher ausführen, geben Sie die Parametergruppe oder eine davon abgeleitete `default.redis3.2` Parametergruppe an, um eine Valkey- oder OSS Redis-Replikationsgruppe (Clustermodus deaktiviert) `default.redis3.2` zu erstellen. Weitere Informationen finden Sie unter [Valkey- und Redis-Parameter OSS](#).

## --Netzwerktyp

Entweder `ipv4`, `ipv` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

## Engine

`redis`

## EngineVersion

`6.0`

Die Namen der Knoten werden vom Namen der Replikationsgruppe abgeleitet, indem `-00#` an den Namen der Replikationsgruppe angehängt wird. Wenn Sie z. B. den Replikationsgruppennamen `myRep1Group` verwenden, lauten die Namen des primären Knotens `myRep1Group-001` und der Read Replicas `myRep1Group-002` bis `myRep1Group-006`.

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `TransitEncryptionEnabled=true` oder `AtRestEncryptionEnabled=true` hinzu und erfüllen die folgenden Bedingungen.

- Auf Ihrer Replikationsgruppe muss Redis Version 3.2.6 oder 4.0.10 OSS ausgeführt werden.
- Die Replikationsgruppe muss in einem Amazon erstellt werden VPC.
- Sie müssen außerdem den Parameter `CacheSubnetGroup` einbinden.
- Sie müssen auch den Parameter `AuthToken` mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH Token (Passwort) angeben, das für die Ausführung von Vorgängen an dieser Replikationsgruppe erforderlich ist.

Der folgende Vorgang erstellt die Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) `myRep1Group` mit drei Knoten, einem primären und zwei Replikaten.

```
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>
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im Thema ElastiCache API [CreateReplicationGroup](#).

## Eine Replikationsgruppe in Valkey oder Redis OSS (Clustermodus aktiviert) von Grund auf neu erstellen

Sie können einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole, der oder der AWS CLI erstellen. ElastiCache API Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) hat 1 bis 500 Shards (API/CLI: Knotengruppen), einen primären Knoten in jedem Shard und bis zu 5 Read Replicas in jedem Shard. Sie können einen Cluster mit einer höheren Anzahl an Shards und einer geringeren Anzahl an Replikaten mit bis zu 90 Knoten pro Cluster erstellen. Diese Clusterkonfiguration reicht von 90 Shards und 0 Replikaten bis hin zu 15 Shards und 5 Replikaten, was dem Höchstwert für die Anzahl erlaubter Replikate entspricht.

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Valkey- oder Redis-Engine-Version 5.0.6 oder höher ist. OSS Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikate pro Shard) und 500 Shards (ein primärer Knoten und keine Replikate) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Zu den häufigsten Fallstricken gehören, dass die Subnetze in der Subnetzgruppe einen zu kleinen CIDR Bereich haben oder dass die Subnetze gemeinsam genutzt und von anderen Clustern häufig genutzt werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

Um eine Erhöhung des Limits zu beantragen, [AWS siehe Service Limits](#) und wählen Sie den Limittyp Nodes per cluster per instance type.

Erstellen eines Clusters in Valkey oder OSS Redis (Clustermodus aktiviert)

- [Erstellen eines Valkey- oder Redis-Clusters OSS \(Clustermodus aktiviert\) \(Konsole\)](#)
- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [Eine Replikationsgruppe in Valkey oder Redis OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen \(\) ElastiCache API](#)

Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) (Konsole)

Informationen zum Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) finden Sie unter. [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Stellen Sie sicher, dass der Cluster-Modus aktiviert ist, Cluster-Modus aktiviert (aufskalieren) und dass mindestens zwei Shards und ein Replikationsknoten in jedem angegeben sind.

Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) von Grund auf neu erstellen (AWS CLI)

Das folgende Verfahren erstellt eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) mithilfe von AWS CLI

Wenn Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und alle ihre Knoten mit einem einzigen Befehlsaufruf. AWS CLI `create-replication-group` Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe.

Für die Benennung von Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus aktiviert) gelten folgende Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

`--replication-group-description`

Beschreibung der Replikationsgruppe.

`--cache-node-type`

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2Amazon-Instanztypen](#).

## --data-tiering-enabled

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

## --cache-parameter-group

Geben Sie die `default.redis6.x.cluster.on` Parametergruppe oder eine davon abgeleitete Parametergruppe an, `default.redis6.x.cluster.on` um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) zu erstellen. Weitere Informationen finden Sie unter [Änderungen der Parameter in Redis OSS 6.x](#).

## --engine

redis

## --engine-version

3.2.4

## --num-node-groups

Die Anzahl der Knotengruppen in dieser Replikationsgruppe. Gültige Werte sind 1 bis 500.

### Note

Das Knoten- oder Shard-Limit kann auf einen Maximalwert von 500 pro Cluster erhöht werden. Um eine Limiterhöhung anzufordern, lesen Sie [AWS Service Limits](#) und wählen Sie den Limittyp "Knoten per Cluster per Instance-Typ".

## --replicas-per-node-group

Die Anzahl der Replikat-Knoten in jeder Knotengruppe. Gültige Werte sind 0 bis 5.

## --Netzwerktyp

Entweder `ipv4`, `ipv6` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `--transit-`

encryption-enabled oder --at-rest-encryption-enabled hinzu und erfüllen die folgenden Bedingungen.

- Auf Ihrer Replikationsgruppe muss Redis OSS Version 3.2.6 oder 4.0.10 ausgeführt werden.
- Die Replikationsgruppe muss in einem Amazon erstellt werden VPC.
- Sie müssen außerdem den Parameter --cache-subnet-group einbinden.
- Sie müssen auch den Parameter --auth-token mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH Token (Passwort) angeben, das für die Ausführung von Vorgängen an dieser Replikationsgruppe erforderlich ist.

Mit dem folgenden Vorgang wird die Replikationsgruppe Valkey oder Redis OSS (Clustermodus aktiviert) `sample-repl-group` mit drei Knotengruppen/Shards (`--num-node-groups`) erstellt, die jeweils drei Knoten, einen primären und zwei Read Replicas (`--`) haben. `replicas-per-node-group`

Für Linux, macOS oder 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>
```

Für 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>
```

Der voranstehende Befehl erzeugt die folgende Ausgabe.

```
{
 "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": {}
 }
}
```

Wenn Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) von Grund auf neu erstellen, können Sie jeden Shard im Cluster mithilfe des `--node-group-configuration` Parameters konfigurieren, wie im folgenden Beispiel gezeigt, der zwei Knotengruppen konfiguriert (Konsole: Shards). Der erste Shard verfügt über zwei Knoten, einen primären Knoten und eine Read Replica. Der zweite Shard verfügt über drei Knoten, einen primären Knoten und zwei Read Replicas.

`--node-group-configuration`

Die Konfiguration jeder Knotengruppe. Der Parameter `--node-group-configuration` besteht aus den folgenden Feldern.

- **PrimaryAvailabilityZone** – Die Availability Zone, in der sich der Primärknoten dieser Knotengruppe befindet. Wenn dieser Parameter weggelassen wird, wird die Availability Zone ElastiCache für den primären Knoten ausgewählt.

Beispiel: `us-west-2a`.

- `ReplicaAvailabilityZones` – Eine durch Kommas getrennte Liste von Availability Zones, in denen sich die Lesereplikate befinden. Die Anzahl der Availability Zones in dieser Liste muss mit dem Wert von `ReplicaCount` übereinstimmen. Wenn dieser Parameter weggelassen wird, werden die ElastiCache Availability Zones für die Replikatknoten ausgewählt.

Beispiel: „us-west-2a,us-west-2b,us-west-2c“

- `ReplicaCount` – Die Anzahl der Replikatknoten in dieser Knotengruppe.
- `Slots` – Eine Zeichenfolge, die den Keyspace für die Knotengruppe angibt. Das Format für die Zeichenfolge ist `startKey-endKey`. Wenn dieser Parameter weggelassen wird, werden ElastiCache die Schlüssel gleichmäßig auf die Knotengruppen verteilt.

Beispiel: "0-4999"

Der folgende Vorgang erstellt die Replikationsgruppe Valkey oder Redis OSS (Clustermodus aktiviert) `new-group` mit zwei Knotengruppen/Shards (`.`). `--num-node-groups` Im Gegensatz zum voranstehenden Beispiel sind die einzelnen Knotengruppen unterschiedlich konfiguriert (`--node-group-configuration`).

Für Linux, macOS oder 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'"
```

Für 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'"
```

Die voranstehende Operation erzeugt die folgende Ausgabe.

```
{
 "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": {}
 }
}
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im Thema AWS CLI [create-replication-group](#).

Eine Replikationsgruppe in Valkey oder Redis OSS (Clustermodus aktiviert) von Grund auf neu erstellen () ElastiCache API

Das folgende Verfahren erstellt eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) mithilfe von. ElastiCache API

Wenn Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und all ihre Knoten mit einem einzigen Aufruf des

Vorgangs. ElastiCache API `CreateReplicationGroup` Schließen Sie die folgenden Parameter ein.

### `ReplicationGroupId`

Der Name der zu erstellenden Replikationsgruppe.

Für die Benennung von Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus aktiviert) gelten folgende Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

### `ReplicationGroupDescription`

Beschreibung der Replikationsgruppe.

### `NumNodeGroups`

Die Anzahl der Knotengruppen, die Sie mit dieser Replikationsgruppe erstellen möchten. Gültige Werte sind 1 bis 500.

### `ReplicasPerNodeGroup`

Die Anzahl der Replikat-Knoten in jeder Knotengruppe. Gültige Werte sind 1 bis 5.

### `NodeGroupConfiguration`

Die Konfiguration jeder Knotengruppe. Der Parameter `NodeGroupConfiguration` besteht aus den folgenden Feldern.

- `PrimaryAvailabilityZone` – Die Availability Zone, in der sich der Primärknoten dieser Knotengruppe befindet. Wenn dieser Parameter weggelassen wird, wird die ElastiCache Availability Zone für den primären Knoten ausgewählt.

Beispiel: `us-west-2a`.

- `ReplicaAvailabilityZones` – Eine Liste der Availability Zones, in denen sich die Lesereplikate befinden. Die Anzahl der Availability Zones in dieser Liste muss mit dem Wert von `ReplicaCount` übereinstimmen. Wenn dieser Parameter weggelassen wird, werden die ElastiCache Availability Zones für die Replikatknoten ausgewählt.
- `ReplicaCount` – Die Anzahl der Replikatknoten in dieser Knotengruppe.

- **Slots** – Eine Zeichenfolge, die den Keyspace für die Knotengruppe angibt. Das Format für die Zeichenfolge ist `startKey-endKey`. Wenn dieser Parameter weggelassen wird, werden ElastiCache die Schlüssel gleichmäßig auf die Knotengruppen verteilt.

Beispiel: "0-4999"

## CacheNodeType

Der Knotentyp für jeden Knoten in der Replikationsgruppe.

ElastiCache unterstützt die folgenden Knotentypen. Im Allgemeinen bieten die Typen der aktuellen Generation im Vergleich zu den entsprechenden Typen der vorherigen Generationen mehr Speicher und Rechenleistung zu niedrigeren Kosten.

Weitere Informationen zu Leistungsdetails für jeden Knotentyp finden Sie unter [EC2Amazon-Instanztypen](#).

### --data-tiering-enabled

Legen Sie diesen Parameter fest, wenn Sie einen R6gd-Knotentyp verwenden. Wenn Sie kein Daten-Tiering verwenden möchten, legen Sie `--no-data-tiering-enabled` fest. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

## CacheParameterGroup

Geben Sie die `default.redis6.x.cluster.on` Parametergruppe oder eine davon abgeleitete Parametergruppe an, `default.redis6.x.cluster.on` um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) zu erstellen. Weitere Informationen finden Sie unter [Änderungen der Parameter in Redis OSS 6.x](#).

### --Netzwerktyp

Entweder `ipv4`, `ipv6` oder `dual-stack`. Wenn Sie Dual-Stack auswählen, müssen Sie den `--IpDiscovery`-Parameter entweder auf `ipv4` oder `ipv6` festlegen.

## Engine

redis

## EngineVersion

6.0

Wenn Sie die Verschlüsselung während der Übertragung oder im Ruhezustand für diese Replikationsgruppe aktivieren möchten, fügen Sie einen oder beide der Parameter `TransitEncryptionEnabled=true` oder `AtRestEncryptionEnabled=true` hinzu und erfüllen die folgenden Bedingungen.

- Auf Ihrer Replikationsgruppe muss Redis OSS Version 3.2.6 oder 4.0.10 ausgeführt werden.
- Die Replikationsgruppe muss in einem Amazon erstellten VPC.
- Sie müssen außerdem den Parameter `CacheSubnetGroup` einbinden.
- Sie müssen auch den Parameter `AuthToken` mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH Token (Passwort) angeben, das für die Ausführung von Vorgängen an dieser Replikationsgruppe erforderlich ist.

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
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>
```

Weitere Informationen und Parameter, die Sie möglicherweise verwenden möchten, finden Sie im ElastiCache API Thema [CreateReplicationGroup](#).

## Anzeigen der Details einer Replikationsgruppe

Unter bestimmten Umständen möchten Sie vielleicht Details zu einer Replikationsgruppe einsehen. Sie können die ElastiCache Konsole, das AWS CLI für ElastiCache oder das verwenden ElastiCache API. Der Konsolenprozess unterscheidet sich für Valkey oder Redis OSS (Clustermodus deaktiviert) und Valkey oder Redis OSS (Clustermodus aktiviert).

## Anzeige der Details einer Replikationsgruppe

- [Valkey oder Redis OSS \(Clustermodus deaktiviert\) mit Replikaten anzeigen](#)
  - [Anzeigen einer Valkey- oder Redis-Replikationsgruppe OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
  - [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) anzeigen \(\)AWS CLI](#)
  - [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) anzeigen \(\)ElastiCache API](#)
- [Eine Replikationsgruppe anzeigen: Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
  - [Anzeigen eines Valkey- oder Redis-Clusters OSS \(Clustermodus aktiviert\) \(Konsole\)](#)
  - [Einen Valkey- oder Redis-Cluster OSS \(Clustermodus aktiviert\) anzeigen \(\)AWS CLI](#)
  - [Einen Valkey- oder Redis-Cluster OSS \(Clustermodus aktiviert\) anzeigen \(\)ElastiCache API](#)
- [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#)
- [Anzeigen der Details einer Replikationsgruppe \(ElastiCache API\)](#)

## Valkey oder Redis OSS (Clustermodus deaktiviert) mit Replikaten anzeigen

Sie können die Details eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mit Replikaten (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole, des for oder des anzeigen. AWS CLI ElastiCache ElastiCache API

## Details eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) anzeigen

- [Anzeigen einer Valkey- oder Redis-Replikationsgruppe OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) anzeigen \(\)AWS CLI](#)
- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) anzeigen \(\)ElastiCache API](#)

## Anzeigen einer Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus deaktiviert) (Konsole)

Informationen zum Anzeigen der Details eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mit Replikaten mithilfe der ElastiCache Konsole finden Sie im Thema. [Details zu Valkey oder Redis OSS \(Clustermodus deaktiviert\) anzeigen \(Konsole\)](#)

Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) anzeigen () AWS CLI

Ein AWS CLI Beispiel, in dem die Details einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) angezeigt werden, finden Sie unter. [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#)

Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) anzeigen () ElastiCache API

Ein ElastiCache API Beispiel, in dem die Details einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) angezeigt werden, finden Sie unter. [Anzeigen der Details einer Replikationsgruppe \(ElastiCache API\)](#)

Eine Replikationsgruppe anzeigen: Valkey oder Redis OSS (Clustermodus aktiviert)

Anzeigen eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) (Konsole)

Informationen zum Anzeigen der Details eines Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert) mithilfe der ElastiCache Konsole finden Sie unter. [Details für einen Valkey- oder Redis-Cluster \(Clustermodus aktiviert OSS\) \(Konsole\) anzeigen](#)

Einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) anzeigen () AWS CLI

Ein ElastiCache CLI Beispiel, in dem die Details einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) angezeigt werden, finden Sie unter. [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#)

Einen Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) anzeigen () ElastiCache API

Ein ElastiCache API Beispiel, in dem die Details einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) angezeigt werden, finden Sie unter. [Anzeigen der Details einer Replikationsgruppe \(ElastiCache API\)](#)

Anzeigen der Details einer Replikationsgruppe (AWS CLI)

Sie können die Details für eine Replikationsgruppe mit dem AWS CLI `describe-replication-groups` Befehl anzeigen. Verfeinern Sie die Auflistung mit den folgenden optionalen Parametern. Wenn die Parameter weggelassen werden, werden Details für bis zu 100 Replikationsgruppen zurückgegeben.

## Optionale Parameter

- `--replication-group-id` – Verwenden Sie diesen Parameter zur Auflistung der Details einer bestimmten Replikationsgruppe. Wenn die angegebene Replikationsgruppe mehr als eine Knotengruppe besitzt, werden die zurückgegebenen Ergebnisse nach Knotengruppe gruppiert.
- `--max-items` – Verwenden Sie diesen Parameter, um die Anzahl der aufgelisteten Replikationsgruppen einzuschränken. Der Wert für `--max-items` darf nicht kleiner als 20 oder größer als 100 sein.

## Example

Der folgende Code listet die Details für bis zu 100 Replikationsgruppen auf.

```
aws elasticache describe-replication-groups
```

Der folgende Code listet die Details für `sample-repl-group` auf.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

Der folgende Code listet die Details für `sample-repl-group` auf.

```
aws elasticache describe-replication-groups --replication-group-id sample-repl-group
```

Der folgende Code listet die Details für bis zu 25 Replikationsgruppen auf.

```
aws elasticache describe-replication-groups --max-items 25
```

Die Ausgabe dieses Vorgangs sollte ungefähr so aussehen (JSONFormat).

```
{
 "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
 }
]
}
```

Weitere Informationen finden Sie im AWS CLI ElastiCache For-Thema [describe-replication-groups](#).

## Anzeigen der Details einer Replikationsgruppe (ElastiCache API)

Sie können die Details für eine Replikation mithilfe des AWS CLI DescribeReplicationGroups Vorgangs anzeigen. Verfeinern Sie die Auflistung mit den folgenden optionalen Parametern. Wenn die Parameter weggelassen werden, werden Details für bis zu 100 Replikationsgruppen zurückgegeben.

### Optionale Parameter

- **ReplicationGroupId** – Verwenden Sie diesen Parameter zur Auflistung der Details einer bestimmten Replikationsgruppe. Wenn die angegebene Replikationsgruppe mehr als eine Knotengruppe besitzt, werden die zurückgegebenen Ergebnisse nach Knotengruppe gruppiert.
- **MaxRecords** – Verwenden Sie diesen Parameter, um die Anzahl der aufgelisteten Replikationsgruppen einzuschränken. Der Wert für MaxRecords darf nicht kleiner als 20 oder größer als 100 sein. Der Standardwert ist 100.

### Example

Der folgende Code listet die Details für bis zu 100 Replikationsgruppen auf.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReplicationGroups
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Der folgende Code listet die Details für myReplGroup auf.

```
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>
```

Der folgende Code listet die Details für bis zu 25 Cluster auf.

```
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>
```

Weitere Informationen finden Sie im ElastiCache API Referenzthema [DescribeReplicationGroups](#).

## Suchen von Endpunkten von Replikationsgruppen

Eine Anwendung kann eine Verbindung zu jedem Knoten in einer Replikationsgruppe herstellen, vorausgesetzt, sie hat den DNS Endpunkt und die Portnummer für diesen Knoten. Je nachdem, ob Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus deaktiviert) oder eine Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus aktiviert) ausführen, sind Sie an unterschiedlichen Endpunkten interessiert.

### Valkey oder Redis (Clustermodus deaktiviert) OSS

Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit Replikaten haben drei Arten von Endpunkten: den primären Endpunkt, den Reader-Endpunkt und die Knotenendpunkte. Der primäre Endpunkt ist ein DNS Name, der immer zum primären Knoten im Cluster aufgelöst wird. Der primäre Endpunkt ist Änderungen an Ihrem Cluster, wie Heraufstufen einer Read Replica in die Rolle des primären Knotens, gegenüber immun. Für Schreibvorgänge empfehlen wir, dass Anwendungen sich ausschließlich mit dem primären Endpunkt verbinden.

Ein Leser-Endpunkt teilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem ElastiCache Cluster auf. Zusätzliche Faktoren, z. B. wenn die Anwendung die Verbindungen erstellt oder wie die Anwendung die Verbindungen (erneut) verwendet, bestimmen die Verteilung des Datenverkehrs. Reader-Endpunkte bleiben hinsichtlich der Cluster-Änderungen beim Hinzufügen oder Entfernen von Replicas in Echtzeit auf dem aktuellen Stand. Sie können die mehreren Read Replicas Ihres ElastiCache (Redis-OSS) Clusters in verschiedenen AWS Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Leser-Endpunkte sicherzustellen.

#### Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS Datensatz, der nach dem Round-Robin-Verfahren in die IP-Adresse eines der Replikatknoten aufgelöst wird.

Für Lesevorgänge können Anwendungen Verbindungen zu jedem Knoten im Cluster herstellen. Im Gegensatz zum primären Endpunkt werden Knotenendpunkte auf bestimmte Endpunkte aufgelöst. Wenn Sie eine Änderung am Cluster vornehmen, wie z. B. Hinzufügen oder Löschen eines Replikats, müssen Sie die Knotenendpunkte in Ihrer Anwendung aktualisieren.

### Valkey oder Redis OSS (Clustermodus aktiviert)

Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mit Replikaten haben eine andere Endpunktstruktur als Valkey- oder Redis-Cluster (Clustermodus deaktiviert), da sie mehrere Shards

(API/CLI: Knotengruppen) haben, was bedeutet, dass sie auch mehrere Primärknoten haben, eine andere Endpunktstruktur haben als Valkey- oder OSS Redis-Cluster (Clustermodus deaktiviert). Valkey oder Redis OSS (Clustermodus aktiviert) haben einen Konfigurationsendpunkt, der alle Primär- und Knotenendpunkte im Cluster „kennt“. Ihre Anwendung stellt eine Verbindung zum Konfigurationsendpunkt her. Immer wenn Ihre Anwendung auf den Konfigurationsendpunkt des Clusters schreibt oder von diesem liest, bestimmen Valkey und Redis hinter den Kulissen OSS, zu welchem Shard der Schlüssel gehört und welcher Endpunkt in diesem Shard verwendet werden soll. Dies läuft alles transparent für Ihre Anwendung ab.

Sie können die Endpunkte für einen Cluster mithilfe der ElastiCache Konsole, der oder der finden.  
AWS CLI ElastiCache API

### Suchen von Replikationsgruppen-Endpunkten

Anweisungen zur Suche nach den Endpunkten für Ihre Replikationsgruppe finden Sie in einem der folgenden Themen:

- [Finden der Endpunkte eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Finden von Endpunkten für einen Valkey- oder Redis-Cluster OSS \(Clustermodus aktiviert\) \(Konsole\)](#)
- [Finden der Endpunkte für Valkey- oder OSS Redis-Replikationsgruppen \(\)AWS CLI](#)
- [Suchen nach Endpunkten für Valkey- oder OSS Redis-Replikationsgruppen \(\) ElastiCache API](#)

## Ändern einer Replikationsgruppe

### Wichtige Einschränkungen

- ElastiCache unterstützt derzeit begrenzte Änderungen einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert), z. B. das Ändern der Engine-Version mithilfe der API Operation `ModifyReplicationGroup` (CLI: `modify-replication-group`). Sie können die Anzahl der Shards (Knotengruppen) in einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mit der API Operation (`:`) ändern. [ModifyReplicationGroupShardConfigurationCLImodify-replication-group-shard-configuration](#) Weitere Informationen finden Sie unter [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#).

Andere Änderungen an einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) erfordern, dass Sie einen Cluster mit dem neuen Cluster erstellen, der die Änderungen enthält.

- Sie können Valkey- oder Redis-Cluster und Replikationsgruppen OSS (Clustermodus deaktiviert) und Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) auf neuere Engine-Versionen aktualisieren. Ein Downgrade auf vorhergehende Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters oder der vorhandenen Replikationsgruppe und der anschließenden Erstellung eines neuen Clusters oder einer neuen Replikationsgruppe möglich. Weitere Informationen finden Sie unter [Versionsverwaltung für ElastiCache](#).
- Sie können einen vorhandenen Cluster ElastiCache mit Valkey oder Redis, der den OSS Clustermodus deaktiviert verwendet, mithilfe der Konsole [ModifyReplicationGroupAPI](#) oder des `modify-replication-group` CLIBefehls auf aktivierten Clustermodus aktualisieren, wie im Beispiel unten gezeigt. Sie können auch den Schritten unter [Ändern des Clustermodus](#) folgen.

Sie können die Einstellungen eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mithilfe der ElastiCache Konsole, der oder der AWS CLI ändern. ElastiCache API ElastiCache unterstützt derzeit eine begrenzte Anzahl von Änderungen an einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert). Andere Änderungen erfordern, dass Sie eine Sicherungskopie der aktuellen Replikationsgruppe erstellen und diese Sicherung dann verwenden, um eine neue Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) zu erstellen.

## Themen

- [Unter Verwendung der AWS Management Console](#)
- [Verwenden des AWS CLI](#)
- [Verwenden des ElastiCache API](#)

### Unter Verwendung der AWS Management Console

Informationen zum Ändern eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) finden Sie unter. [Einen ElastiCache Cluster ändern](#)

### Verwenden des AWS CLI

Im Folgenden finden Sie AWS CLI Beispiele für den `modify-replication-group` Befehl. Sie können mit demselben Befehl andere Änderungen an einer Replikationsgruppe vornehmen.

Aktivieren Sie Multi-AZ für eine bestehende Valkey- oder OSS Redis-Replikationsgruppe:

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --multi-az-enabled = true
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --multi-az-enabled
```

Ändern des Clustermodus von deaktiviert auf aktiviert:

Wenn Sie den Clustermodus von deaktiviert auf aktiviert ändern möchten, müssen Sie zuerst den Clustermodus auf kompatibel festlegen. Im kompatiblen Modus können Ihre Valkey- oder OSS Redis-Clients sowohl im aktivierten als auch im deaktivierten Clustermodus eine Verbindung herstellen. Nachdem Sie alle Valkey- oder OSS Redis-Clients zur Verwendung des aktivierten Clustermodus migriert haben, können Sie die Konfiguration des Clustermodus abschließen und den Clustermodus auf aktiviert setzen.

Für Linux, macOS oder Unix:

Legen Sie den Clustermodus auf kompatibel fest.

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --cache-parameter-group-name myParameterGroupName \
 --cluster-mode compatible
```

Legen Sie den Clustermodus auf aktiviert fest.

```
aws elasticache modify-replication-group \
 --replication-group-id myReplGroup \
 --cluster-mode enabled
```

Für Windows:

Legen Sie den Clustermodus auf kompatibel fest.

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --cache-parameter-group-name myParameterGroupName ^
 --cluster-mode compatible
```

Legen Sie den Clustermodus auf aktiviert fest.

```
aws elasticache modify-replication-group ^
 --replication-group-id myReplGroup ^
 --cluster-mode enabled
```

Weitere Informationen zu diesem AWS CLI `modify-replication-group` Befehl finden Sie unter [modify-replication-group](#) oder [Ändern des Clustermodus](#) im ElastiCache (Redis-OSS) Benutzerhandbuch.

Verwenden des ElastiCache API

Der folgende ElastiCache API Vorgang aktiviert Multi-AZ auf einer vorhandenen Valkey- oder OSS Redis-Replikationsgruppe. Sie können mit derselben Operation andere Änderungen an einer Replikationsgruppe vornehmen.

```
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>
```

Weitere Informationen zu diesem Vorgang finden Sie unter ElastiCache API `ModifyReplicationGroup` [ModifyReplicationGroup](#).

## Löschen einer Replikationsgruppe

Wenn Sie einen Ihrer Cluster mit Replikaten (im API/als Replikationsgruppen bezeichnet CLI) nicht mehr benötigen, können Sie ihn löschen. Wenn Sie eine Replikationsgruppe ElastiCache löschen, werden alle Knoten in dieser Gruppe gelöscht.

Sobald die Operation gestartet wurde, kann sie nicht mehr unterbrochen oder abgebrochen werden.

### Warning

- Wenn Sie einen ElastiCache (Redis-OSS) Cluster löschen, bleiben Ihre manuellen Snapshots erhalten. Sie erhalten außerdem die Möglichkeit, einen letzten Snapshot zu erstellen, bevor der Cluster gelöscht wird. Automatisch erstellte Cache-Snapshots werden nicht aufbewahrt.
- `CreateSnapshot` Zum Erstellen eines endgültigen Snapshots ist eine Genehmigung erforderlich. Ohne diese Genehmigung schlägt der API Aufruf mit einer `Access Denied` Ausnahme fehl.

### Löschen einer Replikationsgruppe (Konsole)

Zum Löschen eines Clusters mit Replikaten informieren Sie sich unter [Löschen eines Clusters in ElastiCache](#).

### Löschen einer Replikationsgruppe (AWS CLI)

Verwenden Sie den Befehl [delete-replication-group](#) um eine Replikationsgruppe zu löschen.

```
aws elasticache delete-replication-group --replication-group-id my-repgroup
```

Sie werden aufgefordert, Ihre Entscheidung zu bestätigen. Geben Sie `y` (Ja) ein, um die Operation sofort zu starten. Einmal gestartet kann der Vorgang nicht mehr rückgängig gemacht werden.

```
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
```

## Löschen einer Replikationsgruppe (ElastiCache API)

Rufen Sie [DeleteReplicationGroup](#) mit dem `ReplicationGroup` Parameter.

### 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

Wenn Sie den Parameter `RetainPrimaryCluster` auf `true` einstellen, werden alle Read Replicas gelöscht, der primäre Cluster wird jedoch beibehalten.

## Ändern der Anzahl von Replikaten

Sie können die Anzahl der Read Replicas in Ihrer Valkey- oder OSS Redis-Replikationsgruppe mithilfe von AWS Management Console, dem oder dem dynamisch erhöhen oder verringern. AWS CLI ElastiCache API Wenn es sich bei Ihrer Replikationsgruppe um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) handelt, können Sie auswählen, mit welchen Shards (Knotengruppen) die Anzahl der Replikate erhöht oder verringert werden soll.

Um die Anzahl der Replikate in Ihrer Replikationsgruppe dynamisch zu ändern, wählen Sie den Vorgang aus der folgenden Tabelle aus, der zu Ihrer Situation passt.

| So gehen Sie vor     | Für Valkey oder Redis OSS (Clustermodus aktiviert)                 | Für Valkey oder Redis OSS (Clustermodus deaktiviert)                                                                                                                       |
|----------------------|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Replikate hinzufügen | <a href="#">Erhöhen der Anzahl der Replikate in einem Shard</a>    | <a href="#">Erhöhen der Anzahl der Replikate in einem Shard</a><br><br><a href="#">Hinzufügen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)</a> |
| Replikate löschen    | <a href="#">Verringern der Anzahl der Replikate in einem Shard</a> | <a href="#">Verringern der Anzahl der Replikate in einem Shard</a><br><br><a href="#">Löschen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)</a> |

## Erhöhen der Anzahl der Replikate in einem Shard

Sie können die Anzahl der Replikate in einer Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus aktiviert) oder einer Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus deaktiviert) auf maximal fünf erhöhen. Sie können dazu den, oder den AWS Management Console verwenden.  
AWS CLI ElastiCache API

### Themen

- [Mit dem AWS Management Console](#)
- [Verwenden von AWS CLI](#)
- [Mit dem ElastiCache API](#)

### Mit dem AWS Management Console

Das folgende Verfahren verwendet die Konsole, um die Anzahl der Replikate in einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) zu erhöhen.

Um die Anzahl der Replikate in Shards zu erhöhen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey oder Redis OSS und dann den Namen der Replikationsgruppe aus, zu der Sie Replikate hinzufügen möchten.
3. Aktivieren Sie das Kontrollkästchen eines jeden Shards, den Sie zu Replikaten hinzufügen möchten.
4. Wählen Sie Add replicas (Replikate hinzufügen).
5. Füllen Sie die Seite Add Replicas to Shards (Replikate zu Shards hinzufügen) aus:
  - Geben Sie für New number of replicas/shard (Neue Anzahl der Replikate pro Shard) die Anzahl der Replikate ein, über die alle Ihre ausgewählten Shards verfügen sollen. Dieser Wert muss größer oder gleich dem Wert für Current Number of Replicas per shard (Aktuelle Anzahl der Replicas pro Shard) und kleiner oder gleich fünf sein. Als funktionelles Minimum empfehlen wir mindestens zwei Replikate.
  - Wählen Sie für Availability Zones entweder Keine Präferenz aus, um für jedes neue Replikat eine Availability Zone ElastiCache ausgewählt zu haben, oder Availability Zones angeben, um für jedes neue Replikat eine Availability Zone auszuwählen.

Geben Sie bei Auswahl von Availability Zones an, für jedes neue Replikat eine Availability Zone aus der Liste an.

6. Klicken Sie auf Add (Hinzufügen), um die Replikate hinzuzufügen, oder auf Cancel (Abbrechen), um den Vorgang abzubrechen.

## Verwenden von AWS CLI

Um die Anzahl der Replikate in einem Valkey- oder OSS Redis-Shard zu erhöhen, verwenden Sie den `increase-replica-count` Befehl mit den folgenden Parametern:

- `--replication-group-id` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl von Replikaten erhöhen möchten.
- `--apply-immediately` oder `--no-apply-immediately` – Erforderlich. Gibt an, ob die Replikatanzahl sofort (`--apply-immediately`) oder während des nächsten Wartungsfensters (`--no-apply-immediately`) erhöht werden soll. `--no-apply-immediately` wird derzeit nicht unterstützt.
- `--new-replica-count` Optional. Gibt die gewünschte Anzahl von Replikatknoten nach der Fertigstellung an, bis zu maximal fünf. Verwenden Sie diesen Parameter für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert), wenn es nur eine Knotengruppe oder Valkey- oder Redis-Gruppe OSS (Clustermodus aktiviert) gibt oder wenn Sie möchten, dass alle Knotengruppen dieselbe Anzahl von Replikaten haben. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- `--replica-configuration` Optional. Ermöglicht Ihnen, die Anzahl der Replikate und Availability Zones für jede Knotengruppe separat festzulegen. Verwenden Sie diesen Parameter für Valkey- oder Redis-Gruppen OSS (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

`--replica-configuration` verfügt über drei optionale Mitglieder:

- `NodeId` Die vierziffrige ID der Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) ist die Shard-ID immer `0001`. Informationen zur ID einer Valkey- oder Redis-Knotengruppe OSS (Shards) finden Sie unter [Auffinden der ID eines Shards](#)
- `NewReplicaCount` Die Anzahl der Replikate, die sich am Ende dieser Operation in dieser Knotengruppe befinden sollen. Der Wert muss höher als die aktuelle Anzahl der Replikate sein

und darf maximal fünf betragen. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.

- **PreferredAvailabilityZones** Eine Liste der **PreferredAvailabilityZone**-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppe befinden sollen. Die Anzahl der **PreferredAvailabilityZone**-Werte muss gleich dem Wert von **NewReplicaCount** plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von weggelassen `--replica-configuration` wird, wählt ElastiCache (RedisOSS) die Availability Zone für jedes der neuen Replikate aus.

#### Important

Sie müssen in Ihren Aufruf entweder den Parameter `--new-replica-count` oder den Parameter `--replica-configuration`, aber nicht beide, einbeziehen.

## Example

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf drei erhöht. Wenn das Beispiel abgeschlossen ist, befinden sich in jeder Knotengruppe drei Replikate. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus deaktiviert) mit einer einzelnen Knotengruppe oder um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus aktiviert) mit mehreren Knotengruppen handelt.

Für Linux, macOS oder Unix:

```
aws elasticache increase-replica-count \
 --replication-group-id sample-repl-group \
 --new-replica-count 3 \
 --apply-immediately
```

Für Windows:

```
aws elasticache increase-replica-count ^
 --replication-group-id sample-repl-group ^
 --new-replica-count 3 ^
 --apply-immediately
```

Im folgenden Beispiel erhöht sich die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf den Wert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

Für Linux, macOS oder 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
```

Für 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
```

Weitere Informationen zur Erhöhung der Anzahl von Replikaten mithilfe von finden Sie [increase-replica-count](#) in der CLI ElastiCache Amazon-Befehlszeilenreferenz.

Mit dem ElastiCache API

Um die Anzahl der Replikate in einem Valkey- oder OSS Redis-Shard zu erhöhen, verwenden Sie die `IncreaseReplicaCount` Aktion mit den folgenden Parametern:

- `ReplicationGroupId` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl von Replikaten erhöhen möchten.

- `ApplyImmediately` – Erforderlich. Gibt an, ob die Replikanzahl sofort (`ApplyImmediately=True`) oder während des nächsten Wartungsfensters (`ApplyImmediately=False`) erhöht werden soll. `ApplyImmediately=False` wird derzeit nicht unterstützt.
- `NewReplicaCount` Optional. Gibt die gewünschte Anzahl von Replikatknoten nach der Fertigstellung an, bis zu maximal fünf. Verwenden Sie diesen Parameter für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert), bei denen es nur eine Knotengruppe gibt, oder für Valkey- oder Redis-Gruppen OSS (Clustermodus aktiviert), bei denen alle Knotengruppen dieselbe Anzahl von Replikaten haben sollen. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- `ReplicaConfiguration` Optional. Ermöglicht Ihnen, die Anzahl der Replikate und Availability Zones für jede Knotengruppe separat festzulegen. Verwenden Sie diesen Parameter für Valkey- oder Redis-Gruppen OSS (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

`ReplicaConfiguration` verfügt über drei optionale Mitglieder:

- `NodeId` Die vierziffrige ID für die Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) ist die Knotengruppen-ID (Shard) immer `0001`. Informationen zur ID einer Valkey- oder Redis-Knotengruppe OSS (Shards) finden Sie unter [Auffinden der ID eines Shards](#)
- `NewReplicaCount` Die Anzahl der Replikate, die sich am Ende dieser Operation in dieser Knotengruppe befinden sollen. Der Wert muss höher als die aktuelle Anzahl der Replikate sein und darf maximal fünf betragen. Wenn dieser Wert nicht größer als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- `PreferredAvailabilityZones` Eine Liste der `PreferredAvailabilityZone`-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppe befinden sollen. Die Anzahl der `PreferredAvailabilityZone`-Werte muss gleich dem Wert von `NewReplicaCount` plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von `ReplicaConfiguration` weggelassen wird, wählt ElastiCache (RedisOSS) die Availability Zone für jedes der neuen Replikate aus.

#### Important

Sie müssen in Ihren Aufruf entweder den Parameter `NewReplicaCount` oder den Parameter `ReplicaConfiguration`, aber nicht beide, einbeziehen.

## Example

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf drei erhöht. Wenn das Beispiel abgeschlossen ist, befinden sich in jeder Knotengruppe drei Replikate. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus deaktiviert) mit einer einzelnen Knotengruppe oder um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus aktiviert) mit mehreren Knotengruppen handelt.

```
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>
```

Im folgenden Beispiel erhöht sich die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf den Wert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

```
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>
```

Weitere Informationen zur Erhöhung der Anzahl von Replikaten mit dem API finden Sie [IncreaseReplicaCount](#) in der [ElastiCache API Amazon-Referenz](#).

## Verringern der Anzahl der Replikate in einem Shard

Sie können die Anzahl der Replikate in einem Shard für Valkey oder Redis OSS (Clustermodus aktiviert) oder in einer Replikationsgruppe für Valkey oder Redis OSS (Clustermodus deaktiviert) verringern:

- Für Valkey oder Redis OSS (Clustermodus deaktiviert) können Sie die Anzahl der Replikate auf eins reduzieren, wenn Multi-AZ aktiviert ist, und auf Null, wenn es nicht aktiviert ist.
- Für Valkey oder Redis OSS (Clustermodus aktiviert) können Sie die Anzahl der Replikate auf Null reduzieren. Es ist jedoch kein Failover auf ein Replikat möglich, wenn Ihr primärer Knoten ausfällt.

Sie können das AWS Management Console, das AWS CLI oder das verwenden, ElastiCache API um die Anzahl der Replikate in einer Knotengruppe (Shard) oder Replikationsgruppe zu verringern.

### Themen

- [Unter Verwendung der AWS Management Console](#)
- [Unter Verwendung des AWS CLI](#)
- [Unter Verwendung der ElastiCache API](#)

### Unter Verwendung der AWS Management Console

Das folgende Verfahren verwendet die Konsole, um die Anzahl der Replikate in einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) zu verringern.

Um die Anzahl der Replikate in einem Valkey- oder Redis-Shard zu verringern OSS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey oder Redis und dann den Namen der Replikationsgruppe ausOSS, aus der Sie Replikate löschen möchten.
3. Aktivieren Sie das Kontrollkästchen eines jeden Shards, aus dem Sie einen Replikatknoten entfernen möchten.
4. Wählen Sie Delete replicas (Replikate löschen).
5. Füllen Sie die Seite Delete Replicas from to Shards (Replikate aus Shards löschen) aus:
  - a. Geben Sie für New number of replicas/shard (Neue Anzahl der Replikate pro Shard) die Anzahl der Replikate ein, über die die ausgewählten Shards verfügen sollen. Dieser Wert

muss größer oder gleich 1 sein. Als funktionelles Minimum empfehlen wir mindestens zwei Replikate pro Shard.

- b. Klicken Sie auf Delete (Löschen), um die Replikate zu löschen, oder auf Cancel (Abbrechen), um den Vorgang abzubrechen.

#### Important

- Wenn Sie die zu löschenden Replikatknoten nicht angeben, wählt ElastiCache (RedisOSS) automatisch die zu löschenden Replikatknoten aus. Dabei versucht ElastiCache (RedisOSS), die Multi-AZ-Architektur für Ihre Replikationsgruppe beizubehalten, gefolgt von der Beibehaltung von Replikaten mit minimaler Replikationsverzögerung gegenüber der Primärgruppe.
- Primäre oder Master-Knoten in einer Replikationsgruppe können nicht gelöscht werden. Wenn Sie einen primären Knoten für die Löschoption angeben, schlägt dies mit einem Fehlerereignis fehl, das darauf hinweist, dass der primäre Knoten zum Löschen ausgewählt wurde.

#### Unter Verwendung des AWS CLI

Um die Anzahl der Replikate in einem Valkey- oder OSS Redis-Shard zu verringern, verwenden Sie den `decrease-replica-count` Befehl mit den folgenden Parametern:

- `--replication-group-id` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl der Replikate verringern möchten.
- `--apply-immediately` oder `--no-apply-immediately` – Erforderlich. Gibt an, ob die Replikatanzahl sofort (`--apply-immediately`) oder während des nächsten Wartungsfensters (`--no-apply-immediately`) verringert werden soll. `--no-apply-immediately` wird derzeit nicht unterstützt.
- `--new-replica-count` Optional. Gibt die Anzahl der von Ihnen gewünschten Replikatknoten an. Der Wert `--new-replica-count` muss ein gültiger Wert kleiner als die aktuelle Anzahl der Replikate in den Knotengruppen sein. Informationen zu den zulässigen Mindestwerten finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#). Wenn der Wert von `--new-replica-count` dieser Anforderung nicht entspricht, schlägt der Aufruf fehl.
- `--replicas-to-remove` Optional. Enthält eine Liste von Knoten, in der die zu entfernenden Replikatknoten IDs angegeben sind.

- `--replica-configuration` Optional. Ermöglicht Ihnen, die Anzahl der Replikate und Availability Zones für jede Knotengruppe separat festzulegen. Verwenden Sie diesen Parameter für Valkey- oder Redis-Gruppen OSS (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

`--replica-configuration` verfügt über drei optionale Mitglieder:

- `NodeId` Die vierziffrige ID der Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) ist die Shard-ID immer `0001`. Informationen zur ID einer Valkey- oder Redis-Knotengruppe OSS (Shards) finden Sie unter [Auffinden der ID eines Shards](#)
- `NewReplicaCount` Ein optionaler Parameter, der die Anzahl der von Ihnen gewünschten Replikatknoten angibt. Der Wert `NewReplicaCount` muss ein gültiger Wert kleiner als die aktuelle Anzahl der Replikate in den Knotengruppen sein. Informationen zu den zulässigen Mindestwerten finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#). Wenn der Wert von `NewReplicaCount` dieser Anforderung nicht entspricht, schlägt der Aufruf fehl.
- `PreferredAvailabilityZones` Eine Liste von `PreferredAvailabilityZone`-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppen befinden. Die Anzahl der `PreferredAvailabilityZone`-Werte muss gleich dem Wert von `NewReplicaCount` plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von weggelassen `--replica-configuration` wird, wählt ElastiCache (RedisOSS) die Availability Zone für jedes der neuen Replikate aus.

#### Important

Sie dürfen nur genau einen der `--new-replica-count`-, `--replicas-to-remove`- oder `--replica-configuration`-Parameter einschließen.

## Example

Im folgenden Beispiel wird mit `--new-replica-count` die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` um eins verringert. Wenn das Beispiel abgeschlossen ist, befindet sich ein Replikat in jeder Knotengruppe. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus deaktiviert) mit einer einzelnen Knotengruppe oder um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus aktiviert) mit mehreren Knotengruppen handelt.

Für Linux, macOS oder Unix:

```
aws elasticache decrease-replica-count
 --replication-group-id sample-repl-group \
 --new-replica-count 1 \
 --apply-immediately
```

Für Windows:

```
aws elasticache decrease-replica-count ^
 --replication-group-id sample-repl-group ^
 --new-replica-count 1 ^
 --apply-immediately
```

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` verringert, indem zwei bestimmte Replikate (`0001` und `0003`) aus der Knotengruppe entfernt werden.

Für Linux, macOS oder Unix:

```
aws elasticache decrease-replica-count \
 --replication-group-id sample-repl-group \
 --replicas-to-remove 0001,0003 \
 --apply-immediately
```

Für Windows:

```
aws elasticache decrease-replica-count ^
 --replication-group-id sample-repl-group ^
 --replicas-to-remove 0001,0003 \
 --apply-immediately
```

Im folgenden Beispiel wird mit `--replica-configuration` die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf den Wert verringert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

Für Linux, macOS oder 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
```

Für 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
```

Weitere Informationen zur Verringerung der Anzahl von Replikaten mithilfe von finden Sie [decrease-replica-count](#) in der CLI ElastiCache Amazon-Befehlszeilenreferenz.

Unter Verwendung der ElastiCache API

Um die Anzahl der Replikate in einem Valkey- oder OSS Redis-Shard zu verringern, verwenden Sie die `DecreaseReplicaCount` Aktion mit den folgenden Parametern:

- `ReplicationGroupId` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl der Replikate verringern möchten.
- `ApplyImmediately` – Erforderlich. Gibt an, ob die Replikatanzahl sofort (`ApplyImmediately=True`) oder während des nächsten Wartungsfensters (`ApplyImmediately=False`) verringert werden soll. `ApplyImmediately=False` wird derzeit nicht unterstützt.
- `NewReplicaCount` Optional. Gibt die Anzahl der von Ihnen gewünschten Replikatknoten an. Der Wert `NewReplicaCount` muss ein gültiger Wert kleiner als die aktuelle Anzahl der Replikate in den Knotengruppen sein. Informationen zu den zulässigen Mindestwerten finden Sie unter [Verringern der Anzahl der Replikate in einem Shard](#). Wenn der Wert von `--new-replica-count` dieser Anforderung nicht entspricht, schlägt der Aufruf fehl.

- `ReplicasToRemove` Optional. Enthält eine Liste von Knoten, die die zu entfernenden Replikatknoten IDs angeben.
- `ReplicaConfiguration` Optional. Enthält eine Liste von Knotengruppen, mit denen Sie die Anzahl der Replikate und Availability Zones separat für jede Knotengruppe festlegen können. Verwenden Sie diesen Parameter für Valkey- oder Redis-Gruppen OSS (Clustermodus aktiviert), bei denen Sie jede Knotengruppe unabhängig konfigurieren möchten.

`ReplicaConfiguration` verfügt über drei optionale Mitglieder:

- `NodeId` Die vierstellige ID für die Knotengruppe, die Sie derzeit konfigurieren. Für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert) ist die Knotengruppen-ID immer `0001`. Informationen zur ID einer Valkey- oder OSS Redis-Knotengruppe (Shards) finden Sie unter [Auffinden der ID eines Shards](#)
- `NewReplicaCount` Die Anzahl der Replikate, die sich am Ende dieser Operation in dieser Knotengruppe befinden sollen. Der Wert muss weniger als die aktuelle Anzahl der Replikate sein. Der Mindestwert ist 1, wenn Multi-AZ aktiviert ist, bzw. 0, wenn Multi-AZ mit Automatic Failover nicht aktiviert ist. Wenn dieser Wert nicht kleiner als die aktuelle Anzahl der Replikate in der Knotengruppe ist, schlägt der Aufruf mit einer Ausnahme fehl.
- `PreferredAvailabilityZones` Eine Liste von `PreferredAvailabilityZone`-Zeichenfolgen, die angeben, in welchen Availability Zones sich die Knoten der Replikationsgruppen befinden. Die Anzahl der `PreferredAvailabilityZone`-Werte muss gleich dem Wert von `NewReplicaCount` plus 1 sein, um dem primären Knoten Rechnung zu tragen. Wenn dieses Mitglied von weggelassen `ReplicaConfiguration` wird, wählt ElastiCache (RedisOSS) die Availability Zone für jedes der neuen Replikate aus.

#### Important

Sie dürfen nur genau einen der `NewReplicaCount`-, `ReplicasToRemove`- oder `ReplicaConfiguration`-Parameter einschließen.

## Example

Im folgenden Beispiel wird mit `NewReplicaCount` die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` um eins verringert. Wenn das Beispiel abgeschlossen ist, befindet sich ein Replikat in jeder Knotengruppe. Diese Zahl gilt unabhängig davon, ob es sich um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus deaktiviert) mit einer einzelnen

Knotengruppe oder um eine Valkey- oder Redis-Gruppe OSS (Cluster-Modus aktiviert) mit mehreren Knotengruppen handelt.

```
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>
```

Im folgenden Beispiel wird die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` verringert, indem zwei bestimmte Replikate (`0001` und `0003`) aus der Knotengruppe entfernt werden.

```
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>
```

Im folgenden Beispiel wird mit `ReplicaConfiguration` die Anzahl der Replikate in der Replikationsgruppe `sample-repl-group` auf den Wert verringert, der für die beiden angegebenen Knotengruppen festgelegt ist. Da es mehrere Knotengruppen gibt, handelt es sich um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten Availability Zones gleich dem Wert von `NewReplicaCount` plus 1 sein. Diese Methode trägt dem primären Knoten für die durch `NodeGroupId` identifizierte Gruppe Rechnung.

```
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>
```

Weitere Informationen zur Verringerung der Anzahl von Replikaten mit dem API finden Sie [DecreaseReplicaCount](#) in der ElastiCache API Amazon-Referenz.

Hinzufügen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)

Die Informationen im folgenden Thema beziehen sich nur auf Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert).

Mit zunehmendem Datenverkehr durch Lesezugriffe empfiehlt es sich, diese Lesezugriffe über mehr Knoten zu verteilen und somit die Belastung durch Lesezugriffe auf jedem Knoten zu reduzieren. In diesem Thema erfahren Sie, wie Sie einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) eine Read Replica hinzufügen.

Eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) kann maximal fünf Read Replicas enthalten. Bei dem Versuch, einer Replikationsgruppe mit bereits fünf Read Replicas ein weiteres Read Replica hinzuzufügen, schlägt die Operation fehl.

Informationen zum Hinzufügen von Replikaten zu einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) finden Sie im Folgenden:

- [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
- [Erhöhen der Anzahl der Replikate in einem Shard](#)

Sie können einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mithilfe der ElastiCache Konsole, der oder der eine Read Replica hinzufügen. AWS CLI ElastiCache API

Verwandte Themen

- [Hinzufügen von Knoten zu einem ElastiCache Cluster](#)
- [Hinzufügen einer Read Replica zu einer Replikationsgruppe \(AWS CLI\)](#)
- [Hinzufügen einer Read Replica zu einer Replikationsgruppe mithilfe von API](#)

Hinzufügen einer Read Replica zu einer Replikationsgruppe (AWS CLI)

Um einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) eine Read Replica hinzuzufügen, verwenden Sie den AWS CLI `create-cache-cluster` Befehl mit dem Parameter, `--replication-group-id` um anzugeben, zu welcher Replikationsgruppe der Cluster (Knoten) hinzugefügt werden soll.

Im folgenden Beispiel wird der Cluster `my-read-replica` erstellt und zur Replikationsgruppe `my-replication-group` hinzugefügt. Die Knotentypen, Parametergruppen, Sicherheitsgruppen, das Wartungsfenster und andere Einstellungen für das Read Replica sind mit denen für die anderen Knoten in `my-replication-group` identisch.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \
 --cache-cluster-id my-read-replica \
 --replication-group-id my-replication-group
```

Für Windows:

```
aws elasticache create-cache-cluster ^
 --cache-cluster-id my-read-replica ^
 --replication-group-id my-replication-group
```

Weitere Informationen zum Hinzufügen einer Read Replica mithilfe von finden Sie unter CLI [create-cache-cluster](#) in der ElastiCache Amazon-Befehlszeilenreferenz.

Hinzufügen einer Read Replica zu einer Replikationsgruppe mithilfe von API

Um eine Read Replica zu einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) hinzuzufügen, verwenden Sie den ElastiCache `CreateCacheCluster` Vorgang mit dem Parameter, `ReplicationGroupId` um anzugeben, zu welcher Replikationsgruppe der Cluster (Knoten) hinzugefügt werden soll.

Im folgenden Beispiel wird der Cluster `myReadReplica` erstellt und zur Replikationsgruppe `myReplicationGroup` hinzugefügt. Die Knotentypen, Parametergruppen, Sicherheitsgruppen, das Wartungsfenster und andere Einstellungen für das Read Replica sind mit denen für die anderen Knoten in `myReplicationGroup` identisch.

```
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>
```

Weitere Informationen zum Hinzufügen einer Read Replica mithilfe von finden Sie unter API [CreateCacheCluster](#) in der ElastiCache APIAmazon-Referenz.

Löschen einer Read Replica für Valkey oder Redis OSS (Cluster-Modus deaktiviert)

Die Informationen im folgenden Thema beziehen sich nur auf Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert).

Wenn sich der Lesedatenverkehr in Ihrer Valkey- oder OSS Redis-Replikationsgruppe ändert, möchten Sie möglicherweise Read Replicas hinzufügen oder entfernen. Das Entfernen eines Knotens aus einer -Replikationsgruppe ist mit dem Löschen eines Clusters vergleichbar, aber mit einigen Einschränkungen:

- Der primäre Knoten kann nicht aus einer Replikationsgruppe entfernt werden. Wenn Sie den primären Knoten entfernen möchten, verfahren Sie wie folgt:

1. Stufen Sie eine Read Replica zum primären Knoten herauf. Weitere Informationen zum Heraufstufen einer Read Replica zum primären Knoten finden Sie unter [Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey- oder Redis-Replikationsgruppen OSS \(Clustermodus deaktiviert\)](#).
  2. Löschen Sie den alten primären Knoten. Eine Einschränkung dieser Methode finden Sie im nächsten Gliederungspunkt.
- Wenn Multi-AZ für eine Replikationsgruppe aktiviert ist, kann das letzte Read Replica nicht aus der Replikationsgruppe entfernt werden. Führen Sie in diesem Fall folgende Schritte aus:
    1. Bearbeiten Sie die Replikationsgruppe, indem Sie Multi-AZ deaktivieren. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).
    2. Löschen Sie das Read Replica.

Sie können eine Read Replica aus einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) entfernen, indem Sie die ElastiCache Konsole, die Option for oder die verwenden. AWS CLI ElastiCache ElastiCache API

Anweisungen zum Löschen eines Clusters aus einer Valkey- oder OSS Redis-Replikationsgruppe finden Sie im Folgenden:

- [Unter Verwendung der AWS Management Console](#)
- [Verwenden Sie den AWS CLI , um einen ElastiCache Cluster zu löschen](#)
- [Mit dem ElastiCache API](#)
- [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)
- [Verringern der Anzahl der Replikate in einem Shard](#)

## Heraufstufen einer Read Replica zu primären Replikationsgruppen für Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert)

Die Informationen im folgenden Thema beziehen sich nur auf Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus deaktiviert).

Sie können eine Valkey- oder Redis-Lesereplik OSS (Clustermodus deaktiviert) mithilfe der AWS Management Console, der oder der zur primären Replikation heraufstufen. AWS CLI ElastiCache API Ein Read Replica kann nicht zum primären Knoten heraufgestuft werden, wenn für die - Replikationsgruppe Multi-AZ mit automatischem Failover aktiviert ist. Gehen Sie wie folgt vor, um ein Valkey- oder Redis-Replikat OSS (Clustermodus deaktiviert) in einer Multi-AZ-fähigen Replikationsgruppe zum Primärreplikat hochzustufen:

1. Bearbeiten Sie die Replikationsgruppe so, dass Multi-AZ deaktiviert wird (dazu müssen sich nicht alle Ihre Cluster in derselben Availability Zone befinden). Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).
2. Stufen Sie das Lesereplikat zum primären Knoten herauf.
3. Bearbeiten Sie die Replikationsgruppe so, dass Multi-AZ wieder aktiviert ist.

Multi-AZ ist für Replikationsgruppen, auf denen Redis 2.6.13 oder früher ausgeführt wird, nicht verfügbar. OSS

### Verwenden von AWS Management Console

Im folgenden Verfahren wird ein Replikatknoten mithilfe der Konsole zum primären Knoten heraufgestuft.

### Heraufstufen eines Lesereplikats zur primären (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wenn das Replikat, das Sie heraufstufen möchten, Mitglied einer Valkey- oder Redis-Replikationsgruppe OSS (deaktivierter Clustermodus) ist, in der Multi-AZ aktiviert ist, ändern Sie die Replikationsgruppe so, dass Multi-AZ deaktiviert ist, bevor Sie fortfahren. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).
3. Wählen Sie Valkey oder Redis OSS aus und wählen Sie dann aus der Clusterliste die Replikationsgruppe aus, die Sie ändern möchten. Diese Replikationsgruppe muss die "Redis"-

- Engine und nicht die "Clustered Redis"-Engine ausführen und muss aus mindestens zwei Knoten bestehen.
- Wählen Sie aus der Liste der Knoten den Replikationsknoten aus, den Sie zum primären Knoten heraufstufen möchten. Wählen Sie dann für Actions (Aktionen) die Option Promote (Heraufstufen) aus.
  - Führen Sie im Dialogfeld Promote Read Replica (Read Replica heraufstufen) die folgenden Schritte aus:
    - Wählen Sie für Apply Immediately (Sofort anwenden) die Option Yes (Ja) aus, wenn das Read Replica sofort heraufgestuft werden soll, oder die Option No (Nein), um es im nächsten Wartungsfenster des Clusters heraufzustufen.
    - Wählen Sie Promote, um die Read Replica heraufzustufen, oder Cancel, um die Operation abubrechen.
  - Wenn für den Cluster vor Beginn des Hochstufens Multi-AZ aktiviert war, warten Sie, bis der Status der Replikationsgruppe available (verfügbar) lautet, bevor Sie den Cluster so bearbeiten, dass Multi-AZ wieder aktiviert wird. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

Verwenden Sie den AWS CLI

Sie können eine Read Replica nicht zum primären Knoten heraufstufen, wenn für die Replikationsgruppe Multi-AZ aktiviert ist. In einigen Fällen ist das heraufzustufende Replikat möglicherweise Teil einer Replikationsgruppe, in der Multi-AZ aktiviert ist. Sollte dies der Fall sein, muss Multi-AZ vor dem Fortfahren für die Replikationsgruppe deaktiviert werden. Für dieses Verfahren müssen sich nicht alle Ihre Cluster in der gleichen Availability Zone befinden. Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie unter [Ändern einer Replikationsgruppe](#).

Mit dem folgenden AWS CLI Befehl wird die Replikationsgruppe geändert `sample-repl-group`, sodass die Read Replica zur primären Replica in `my-replica-1` der Replikationsgruppe wird.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id sample-repl-group \
 --primary-cluster-id my-replica-1
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id sample-repl-group ^
 --primary-cluster-id my-replica-1
```

Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie unter [modify-replication-group](#) in der ElastiCache Amazon-Befehlszeilenreferenz.

Mit dem ElastiCache API

Sie können eine Read Replica nicht zum primären Knoten heraufstufen, wenn für die Replikationsgruppe Multi-AZ aktiviert ist. In einigen Fällen ist das heraufzustufende Replikat möglicherweise Teil einer Replikationsgruppe, in der Multi-AZ aktiviert ist. Sollte dies der Fall sein, muss Multi-AZ vor dem Fortfahren für die Replikationsgruppe deaktiviert werden. Für dieses Verfahren müssen sich nicht alle Ihre Cluster in der gleichen Availability Zone befinden. Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie unter [Ändern einer Replikationsgruppe](#).

Mit der folgenden ElastiCache API Aktion wird die Replikationsgruppe geändert `myReplGroup`, sodass die Read Replica zur primären Replica in `myReplica-1` der Replikationsgruppe wird.

```
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>
```

Weitere Informationen zum Ändern einer Replikationsgruppe finden Sie unter [ModifyReplicationGroup](#) in der ElastiCache API Amazon-Referenz.

## Verwaltung der ElastiCache Cluster-Wartung

Jeder Cluster verfügt über ein wöchentliches Wartungsfenster, während dem alle Systemänderungen angewendet werden. Bei Valkey und Redis OSS haben Replikationsgruppen dasselbe wöchentliche Wartungsfenster. Wenn Sie bei der Erstellung oder Änderung eines Clusters oder einer Replikationsgruppe kein bevorzugtes Wartungsfenster angeben, ElastiCache weisen Sie innerhalb des Wartungsfensters Ihrer Region an einem zufällig ausgewählten Wochentag ein 60-minütiges Wartungsfenster zu.

Das 60-minütige Wartungsfenster wird zufällig aus einem 8-Stunden-Zeitraum pro Region ausgewählt. Die folgende Tabelle listet die Blöcke für jede Region auf, von denen die Standard-Wartungsfenster zugewiesen werden. Sie können ein bevorzugtes Wartungsfenster außerhalb des Wartungsfensterblocks der Region auswählen.

| Regionscode    | Name der Region                 | Regionale Wartungsfenster |
|----------------|---------------------------------|---------------------------|
| ap-northeast-1 | Region Asien-Pazifik (Tokio)    | 13:00 — 21:00 Uhr UTC     |
| ap-northeast-2 | Region Asien-Pazifik (Seoul)    | 12:00 — 20:00 Uhr UTC     |
| ap-northeast-3 | Region Asien-Pazifik (Osaka)    | 12:00 — 20:00 Uhr UTC     |
| ap-southeast-3 | Region Asien-Pazifik (Jakarta)  | 14:00 — 22:00 UTC         |
| ap-south-1     | Region Asien-Pazifik (Mumbai)   | 17:30 — 1:30 UTC          |
| ap-southeast-1 | Region Asien-Pazifik (Singapur) | 14:00 — 22:00 UTC         |
| cn-north-1     | Region China (Peking)           | 14:00 — 22:00 UTC         |
| cn-northwest-1 | Region China (Ningxia)          | 14:00 — 22:00 UTC         |
| ap-east-1      | Region Asien-Pazifik (Hongkong) | 13:00 — 21:00 Uhr UTC     |
| ap-southeast-2 | Region Asien-Pazifik (Sydney)   | 12:00 — 20:00 Uhr UTC     |
| eu-west-3      | Region Europa (Paris)           | 23:59 — 07:29 UTC         |
| af-south-1     | Region Afrika (Kapstadt)        | 13:00 — 21:00 Uhr UTC     |

| Regionscode   | Name der Region                | Regionale Wartungsfenster |
|---------------|--------------------------------|---------------------------|
| eu-central-1  | Region Europa (Frankfurt)      | 23:00 — 07:00 UTC         |
| eu-west-1     | Region Europa (Irland)         | 22:00 — 06:00 UTC         |
| eu-west-2     | Region Europa (London)         | 23:00 — 07:00 UTC         |
| me-south-1    | Region Naher Osten (Bahrain)   | 13:00 — 21:00 UTC         |
| me-central-1  | Region Naher Osten (UAE)       | 13:00 — 21:00 Uhr UTC     |
| eu-south-1    | Region Europa (Mailand)        | 21:00 — 05:00 UTC         |
| sa-east-1     | Region Südamerika (São Paulo)  | 01:00 — 09:00 UTC         |
| us-east-1     | Region USA Ost (Nord-Virginia) | 03:00 — 11:00 UTC         |
| us-east-2     | Region USA Ost (Ohio)          | 04:00 — 12:00 UTC         |
| us-gov-west-1 | AWS GovCloud (US) Region       | 06:00 — 14:00 Uhr UTC     |
| us-west-1     | Region US West (N. California) | 06:00 — 14:00 UTC         |
| us-west-2     | Region USA West (Oregon)       | 06:00 — 14:00 UTC         |

## Das Wartungsfenster Ihres Clusters oder Ihrer Replikationsgruppe ändern

Das Wartungsfenster sollte in den Zeitraum mit der geringsten Nutzung fallen und daher unter Umständen von Zeit zu Zeit geändert werden. Sie können den Cluster oder die Replikationsgruppe ändern und einen Zeitraum mit einer Dauer von bis zu 24 Stunden festlegen, in dem alle angeforderten Wartungsaktivitäten durchgeführt werden sollen. In diesem Zeitraum werden alle verzögerten oder ausstehenden Cluster-Änderungen, die Sie angefordert haben, ausgeführt.

### Note

Wenn Sie Änderungen am Knotentyp und/oder Engine-Upgrades sofort anwenden möchten, AWS Management Console klicken Sie auf das Feld Jetzt anwenden. Andernfalls werden diese Änderungen während des nächsten geplanten Wartungsfensters angewendet.

Informationen zur Verwendung des API finden Sie unter [modify-replication-group](#) oder [modify-cache-cluster](#).

## Weitere Informationen

Informationen zu Ihrem Wartungsfenster und dem Austausch von Knoten finden Sie unter:

- [ElastiCache Wartung](#) — FAQ bei Wartung und Austausch von Knoten
- [Knoten ersetzen \(Memcached\)](#)— Verwaltung des Knotenaustauschs für Memcached
- [Einen ElastiCache Cluster ändern](#) – Ändern des Wartungsfensters für einen Cluster
- [Knoten ersetzen \(Valkey und OSS Redis\)](#) – Verwalten des Knotenaustauschs
- [Ändern einer Replikationsgruppe](#) – Ändern des Wartungsfensters einer Replikationsgruppe

## Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen

Amazon ElastiCache verwendet Parameter, um die Laufzeiteigenschaften Ihrer Knoten und Cluster zu steuern. In der Regel enthalten neuere Engine-Versionen zusätzliche Parameter zur Unterstützung der neueren Funktionalität. Tabellen mit Memcached-Parametern finden Sie unter. [Memcached-spezifische Parameter](#) Tabellen mit Valkey- und OSS Redis-Parametern finden Sie unter. [Valkey- und Redis-Parameter OSS](#)

Wie zu erwarten, werden einige Parameterwerte, wie z. B. `maxmemory` durch die Engine und den Knotentyp bestimmt. Eine Tabelle dieser Memcached-Parameterwerte nach Knotentyp finden Sie unter. [Spezifische Parameter des Memcached-Knotentyps](#) Eine Tabelle dieser Valkey- und OSS Redis-Parameterwerte nach Knotentyp finden Sie unter. [OSSRedis-Knotentyp-spezifische Parameter](#)

### Note

Eine Liste der Memcached-spezifischen Parameter finden Sie unter [Memcached-spezifische Parameter](#).

## Themen

- [Parameterverwaltung in ElastiCache](#)

- [Cache-Parametergruppenstufen in ElastiCache](#)
- [Eine ElastiCache Parametergruppe erstellen](#)
- [ElastiCache Parametergruppen nach Namen auflisten](#)
- [Werte einer ElastiCache Parametergruppe auflisten](#)
- [Änderung einer ElastiCache Parametergruppe](#)
- [Löschen einer ElastiCache Parametergruppe](#)
- [Motorspezifische Parameter](#)

## Parameterverwaltung in ElastiCache

ElastiCache Parameter werden zur einfacheren Parameterverwaltung in benannten Parametergruppen zusammengefasst. Eine Parametergruppe stellt eine Kombination spezifischer Werte für die Parameter dar, die der Engine-Software während des Systemstarts übergeben werden. Diese Werte bestimmen, wie sich die Engine-Prozesse auf jedem Knoten während der Laufzeit verhalten. Die Parameterwerte für eine bestimmte Parametergruppe gelten für alle Knoten, die der betreffenden Gruppe zugehören. Dabei spielt es keine Rolle, zu welchem Cluster sie gehören.

Zur Feinabstimmung der Leistung Ihres Clusters können Sie einige Parameterwerte bearbeiten oder die Parametergruppe des Clusters ändern.

- Standard-Parametergruppen können nicht geändert oder gelöscht werden. Wenn Sie benutzerdefinierte Parameterwerte benötigen, müssen Sie eine benutzerdefinierte Parametergruppe erstellen.
- Für Memcached müssen die Parametergruppenfamilie und der Cluster, dem Sie sie zuweisen, kompatibel sein. Wenn Ihr Cluster z. B. Memcached Version 1.4.8 ausführt, können Sie nur Parametergruppen, Standard oder benutzerdefiniert, von der Memcached 1.4-Familie verwenden.

Für Redis müssen die Parametergruppenfamilie und der Cluster, dem Sie sie zuweisen, kompatibel sein. Wenn auf Ihrem Cluster beispielsweise Redis OSS Version 3.2.10 ausgeführt wird, können Sie nur Standardparametergruppen oder benutzerdefinierte Parametergruppen aus der Redis 3.2-Familie verwenden. OSS

- Wenn Sie die Parametergruppe eines Clusters ändern, müssen die Werte aller bedingungsabhängig veränderbaren Parameter in der aktuellen und in der neuen Parametergruppe übereinstimmen.
- Wenn Sie bei Memcached die Parameter eines Clusters ändern, wird die Änderung sofort auf den Cluster angewendet. Dies gilt unabhängig davon, ob Sie die Parametergruppe des Clusters oder einen Parameterwert innerhalb der Parametergruppe des Clusters ändern. Anhand der Spalte Changes Take Effect (Änderungen treten in Kraft) in den Tabellen für [Memcached-spezifische Parameter](#) können Sie bestimmen, wann eine bestimmte Parameteränderung übernommen wird. Weitere Informationen zum Neustarten der Knoten eines Clusters finden Sie unter [Neustarten von Clustern](#).
- Wenn Sie bei Redis die Parameter eines Clusters ändern, wird die Änderung entweder sofort oder, mit den unten genannten Ausnahmen, nach dem Neustart der Clusterknoten auf den Cluster angewendet. Dies gilt unabhängig davon, ob Sie die Parametergruppe des Clusters oder einen Parameterwert innerhalb der Parametergruppe des Clusters ändern. Anhand der Spalte Changes

Take Effect (Änderungen treten in Kraft) in den Tabellen für [Valkey- und Redis-Parameter OSS](#) können Sie bestimmen, wann eine bestimmte Parameteränderung übernommen wird.

Weitere Informationen zum Neustarten von Valkey- oder Redis-Knoten finden Sie unter [OSS Knoten neu starten](#)

**i** Änderungen der Parameter Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
  - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
  2. Löschen Sie den Cluster. Siehe [Löschen von Clustern](#).
  3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

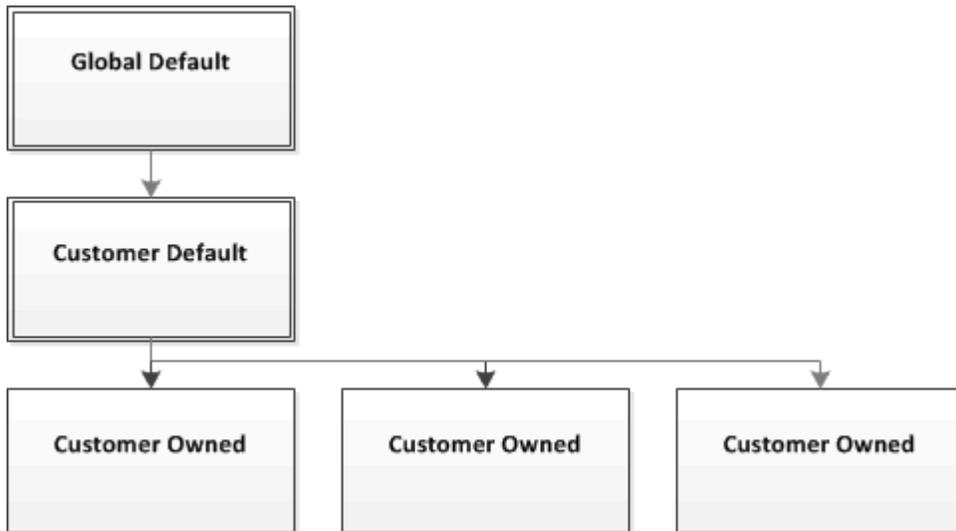
- Sie können Parametergruppen globalen Datenspeichern von Valkey und Redis zuordnen. OSS Globale Datenspeicher sind eine Sammlung von einem oder mehreren Clustern, die sich über Regionen erstrecken. AWS In diesem Fall wird die Parametergruppe von allen Clustern gemeinsam genutzt, die den globalen Datenspeicher bilden. Alle Änderungen an der Parametergruppe des primären Clusters werden auf alle verbleibenden Cluster im globalen Datenspeicher repliziert. Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#).

Sie können überprüfen, ob eine Parametergruppe Teil eines globalen Datenspeichers ist, indem Sie an folgenden Stellen suchen:

- In der ElastiCache Konsole auf der Seite „Parametergruppen“ das globale Attribut Ja/Nein
- Die Ja/Nein-Eigenschaft der Operation IsGlobal [CacheParameterGroupAPI](#)

## Cache-Parametergruppenstufen in ElastiCache

Amazon ElastiCache hat drei Stufen von Cache-Parametergruppen, wie im Folgenden dargestellt.



### ElastiCache Amazon-Parametergruppenstufen

#### Global Default

Die Stammparametergruppe der obersten Ebene für alle ElastiCache Amazon-Kunden in der Region.

Die globale Standard-Cache-Parametergruppe:

- Ist für den Kunden reserviert ElastiCache und steht ihm nicht zur Verfügung.

#### Customer Default

Eine Kopie der Cache-Parametergruppe „Global Default“, die für die Verwendung durch den Kunden erstellt wird.

Die Cache-Parametergruppe „Customer Default“:

- Ist erstellt und gehört ihm ElastiCache.
- Ist für den Kunden zur Verwendung als Cache-Parametergruppe für alle Cluster verfügbar, die eine Engine-Version ausführen, die von dieser Cache-Parametergruppe unterstützt wird.
- Kann vom Kunden nicht bearbeitet werden.

#### Customer Owned

Eine Kopie der Cache-Parametergruppe „Customer Default“. Die Cache-Parametergruppe „Customer Owned“ wird immer dann erstellt, wenn der Kunde eine Cache-Parametergruppe anlegt.

Die Cache-Parametergruppe „Customer Owned“:

- Wird vom Kunden erstellt und ist dessen Eigentum.
- Kann beliebigen kompatiblen Clustern des Kunden zugeordnet werden.
- Kann vom Kunden in eine benutzerdefinierte Cache-Parametergruppe abgewandelt werden.

Nicht alle Parameterwerte können geändert werden. Weitere Informationen zu Memcached-Werten finden Sie unter. [Memcached-spezifische Parameter](#) Weitere Informationen zu Valkey- und OSS Redis-Werten finden Sie unter. [Valkey- und Redis-Parameter OSS](#)

## Eine ElastiCache Parametergruppe erstellen

Sie müssen eine neue Parametergruppe erstellen, wenn Sie die Standardwerte für einen oder mehrere Parameterwerte ändern möchten. Sie können eine Parametergruppe mithilfe der ElastiCache Konsole AWS CLI, der oder der erstellen ElastiCache API.

### Eine ElastiCache Parametergruppe erstellen (Konsole)

Das folgende Verfahren zeigt, wie Sie eine Parametergruppe mithilfe der ElastiCache Konsole erstellen.

Um eine Parametergruppe mit der ElastiCache Konsole zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Um eine Parametergruppe zu erstellen, wählen Sie Create Parameter Group.

Der Bildschirm Create Parameter Group wird angezeigt.

4. Wählen Sie aus der Liste Family die Familie der Parametergruppe aus, die als Vorlage für Ihre Parametergruppe dienen soll.

Die Parametergruppenfamilie, wie z. B. memcached1.4 oder redis3.2, definiert die tatsächlichen Parameter in Ihrer Parametergruppe und deren Anfangswerte. Die Familie der Parametergruppe muss mit der Engine und Version des Clusters übereinstimmen.

5. Geben Sie in das Feld Name einen eindeutigen Namen für diese Parametergruppe ein.

Wenn Sie einen Cluster erstellen oder die Parametergruppe eines Clusters ändern, wählen Sie die Parametergruppe nach Namen aus. Daher wird empfohlen, einen informativen und die Familie der Parametergruppe identifizierenden Namen zu wählen.

Namenseinschränkungen für die Parametergruppe:

- Muss mit einem Buchstaben beginnen. ASCII
  - Kann nur ASCII Buchstaben, Ziffern und Bindestriche enthalten.
  - Er muss 1 – 255 Zeichen enthalten.
  - Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
  - Er darf nicht mit einem Bindestrich enden.
6. Geben Sie in das Feld Description eine Beschreibung für die Parametergruppe ein.
  7. Um die Parametergruppe zu erstellen, wählen Sie Create.

Um den Vorgang zu beenden, ohne die Parametergruppe zu erstellen, wählen Sie Cancel.

8. Die erstellte Parametergruppe besitzt die Standardwerte der Familie. Zum Ändern der Standardwerte müssen Sie die Parametergruppe abändern. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

## Eine ElastiCache Parametergruppe erstellen (AWS CLI)

Um eine Parametergruppe mit dem zu erstellen AWS CLI, verwenden Sie den Befehl `create-cache-parameter-group` mit diesen Parametern.

- `--cache-parameter-group-name` Der Name der Parametergruppe.

Namenseinschränkungen für die Parametergruppe:

- Muss mit einem ASCII Buchstaben beginnen.
  - Kann nur ASCII Buchstaben, Ziffern und Bindestriche enthalten.
  - Er muss 1 – 255 Zeichen enthalten.
  - Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
  - Er darf nicht mit einem Bindestrich enden.
- `--cache-parameter-group-family` Die Engine und Versionsfamilie der Parametergruppe.
  - `--description` Eine vom Benutzer eingegebene Beschreibung der Parametergruppe.

## Example

Im folgenden Beispiel wird eine Parametergruppe mit dem Namen `myMem14` erstellt, wobei die Familie `memcached1.4` als Vorlage verwendet wird.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name myMem14 \
 --cache-parameter-group-family memcached1.4 \
 --description "My first parameter group"
```

Für Windows:

```
aws elasticache create-cache-parameter-group ^
 --cache-parameter-group-name myMem14 ^
 --cache-parameter-group-family memcached1.4 ^
 --description "My first parameter group"
```

Die Ausgabe dieses Befehls sollte in etwa folgendermaßen aussehen.

```
{
 "CacheParameterGroup": {
 "CacheParameterGroupName": "myMem14",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
}
```

## Example

Im folgenden Beispiel wird eine Parametergruppe mit dem Namen `myRed28` erstellt, wobei die Familie `redis2.8` als Vorlage verwendet wird.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name myRed28 \
 --cache-parameter-group-family redis2.8 \
 --description "My first parameter group"
```

Für Windows:

```
aws elasticache create-cache-parameter-group ^
 --cache-parameter-group-name myRed28 ^
 --cache-parameter-group-family redis2.8 ^
 --description "My first parameter group"
```

Die Ausgabe dieses Befehls sollte in etwa folgendermaßen aussehen.

```
{
 "CacheParameterGroup": {
 "CacheParameterGroupName": "myRed28",
 "CacheParameterGroupFamily": "redis2.8",
 "Description": "My first parameter group"
 }
}
```

Die erstellte Parametergruppe besitzt die Standardwerte der Familie. Zum Ändern der Standardwerte müssen Sie die Parametergruppe abändern. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

Weitere Informationen finden Sie unter [create-cache-parameter-group](#).

Eine ElastiCache Parametergruppe () erstellen ElastiCache API

Um eine Parametergruppe mit dem zu erstellen ElastiCache API, verwenden Sie die `CreateCacheParameterGroup` Aktion mit diesen Parametern.

- `ParameterGroupName` Der Name der Parametergruppe.

Namenseinschränkungen für die Parametergruppe:

- Muss mit einem ASCII Buchstaben beginnen.
- Kann nur ASCII Buchstaben, Ziffern und Bindestriche enthalten.
- Er muss 1 – 255 Zeichen enthalten.
- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.
- `CacheParameterGroupFamily` Die Engine und Versionsfamilie der Parametergruppe. Beispiel, `memcached1.4`.
- `CacheParameterGroupFamily` Die Engine und Versionsfamilie der Parametergruppe. Beispiel, `redis2.8`.

- **Description** Eine vom Benutzer eingegebene Beschreibung der Parametergruppe.

## Example

Im folgenden Beispiel wird eine Parametergruppe mit dem Namen myMem14 erstellt, wobei die Familie memcached1.4 als Vorlage verwendet wird.

```
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>
```

Die Antwort auf diese Aktion sollte in etwa folgendermaßen aussehen.

```
<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

Im folgenden Beispiel wird eine Parametergruppe mit dem Namen myRed28 erstellt, wobei die Familie redis2.8 als Vorlage verwendet wird.

```
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>
```

Die Antwort auf diese Aktion sollte in etwa folgendermaßen aussehen.

```
<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>
```

Die erstellte Parametergruppe besitzt die Standardwerte der Familie. Zum Ändern der Standardwerte müssen Sie die Parametergruppe abändern. Weitere Informationen finden Sie unter [Änderung einer ElastiCache Parametergruppe](#).

Weitere Informationen finden Sie unter [CreateCacheParameterGroup](#).

## ElastiCache Parametergruppen nach Namen auflisten

Sie können die Parametergruppen mithilfe der ElastiCache Konsole AWS CLI, der oder der auflisten ElastiCacheAPI.

### Auflisten von Parametergruppen nach Namen (Konsole)

Das folgende Verfahren zeigt, wie Sie mit der ElastiCache Konsole eine Liste der Parametergruppen anzeigen können.

Um Parametergruppen mithilfe der ElastiCache Konsole aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.

### ElastiCache Parametergruppen nach Namen auflisten (AWS CLI)

Verwenden Sie den Befehl, um mit dem AWS CLI eine Liste von Parametergruppen zu generierend `describe-cache-parameter-groups`. Wenn Sie den Namen einer Parametergruppe angeben, wird nur die betreffende Parametergruppe aufgelistet. Wenn Sie keinen Namen einer Parametergruppe angeben, werden bis zu `--max-records` Parametergruppen aufgelistet. In beiden Fälle werden Name, Familie und Beschreibung der Parametergruppe aufgelistet.

### Example

Der folgende Beispielcode listet die Parametergruppe `myMem14` auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myMem14
```

Für Windows:

```
aws elasticache describe-cache-parameter-groups ^
 --cache-parameter-group-name myMem14
```

Die Ausgabe dieses Befehls sieht folgendermaßen aus und listet Name, Familie und Beschreibung der Parametergruppe auf.

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myMem14",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
]
}
```

## Example

Der folgende Beispielcode listet die Parametergruppe myRed28 auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myRed28
```

Für Windows:

```
aws elasticache describe-cache-parameter-groups ^
 --cache-parameter-group-name myRed28
```

Die Ausgabe dieses Befehls sieht folgendermaßen aus und listet Name, Familie und Beschreibung der Parametergruppe auf.

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myRed28",
 "CacheParameterGroupFamily": "redis2.8",
 "Description": "My first parameter group"
 }
]
}
```

## Example

Der folgende Beispielcode listet die Parametergruppe myRed56 für Parametergruppen auf, die auf der OSS Redis-Engine-Version 5.0.6 und höher ausgeführt werden. Wenn die Parametergruppe

Teil einer [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#) ist, ist der in der Ausgabe zurückgegebene `IsGlobal`-Eigenschaftswert `Yes`.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myRed56
```

Für Windows:

```
aws elasticache describe-cache-parameter-groups ^
 --cache-parameter-group-name myRed56
```

Die Ausgabe dieses Befehls sieht ungefähr so aus und listet den Namen, die Familie `isGlobal` und die Beschreibung für die Parametergruppe auf.

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myRed56",
 "CacheParameterGroupFamily": "redis5.0",
 "Description": "My first parameter group",
 "IsGlobal": "yes"
 }
]
}
```

## Example

Der folgenden Beispielcode listet bis zu 10 Parametergruppen auf.

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

Die JSON Ausgabe dieses Befehls sieht ungefähr so aus und listet für jede Parametergruppe den Namen, die Familie, die Beschreibung auf und, im Fall von `redis5.6`, ob die Parametergruppe Teil eines globalen Datenspeichers (`isGlobal`) ist.

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

Weitere Informationen finden Sie unter [describe-cache-parameter-groups](#).

## ElastiCache Parametergruppen nach Namen () auflisten ElastiCache API

Verwenden Sie die `DescribeCacheParameterGroups` Aktion, um mit dem ElastiCache API eine Liste von Parametergruppen zu generieren. Wenn Sie den Namen einer Parametergruppe angeben, wird nur die betreffende Parametergruppe aufgelistet. Wenn Sie keinen Namen einer Parametergruppe angeben, werden bis zu `MaxRecords` Parametergruppen aufgelistet. In beiden Fälle werden Name, Familie und Beschreibung der Parametergruppe aufgelistet.

### Example

Der folgende Beispielcode listet die Parametergruppe `myMem14` auf.

```
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>
```

Die Antwort auf diese Aktion sieht folgendermaßen aus und listet Name, Familie und Beschreibung jeder Parametergruppe auf.

```
<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

Der folgenden Beispielcode listet bis zu 10 Parametergruppen auf.

```
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>
```

Die Antwort auf diese Aktion sieht ungefähr so aus und listet den Namen, die Familie, die Beschreibung und, im Fall von redis5.6, falls die Parametergruppe zu einem globalen Datenspeicher (isGlobal) gehört, für jede Parametergruppe auf.

```
<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

Der folgende Beispielcode listet die Parametergruppe 28 auf. 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>
```

Die Antwort auf diese Aktion sieht folgendermaßen aus und listet den Namen, die Familie und die Beschreibung auf.

```
<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

Der folgende Beispielcode listet die Parametergruppe myRed56 auf.

```
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>
```

Die Antwort auf diese Aktion sieht ungefähr so aus und listet den Namen, die Familie, die Beschreibung auf und ob die Parametergruppe Teil eines globalen Datenspeichers ist (`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>
```

Weitere Informationen finden Sie unter [DescribeCacheParameterGroups](#).

## Werte einer ElastiCache Parametergruppe auflisten

Sie können die Parameter und ihre Werte für eine Parametergruppe mithilfe der ElastiCache Konsole, AWS CLI, oder der auflisten ElastiCache API.

### Die Werte einer ElastiCache Parametergruppe auflisten (Konsole)

Das folgende Verfahren zeigt, wie Sie die Parameter und ihre Werte für eine Parametergruppe mithilfe der ElastiCache Konsole auflisten.

Um die Parameter einer Parametergruppe und ihre Werte mithilfe der ElastiCache Konsole aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Wählen Sie die Parametergruppe aus, für die Sie die Parameter und Werte auflisten möchten, indem Sie das Kontrollkästchen links neben dem Namen der Parametergruppe aktivieren.

Die Parameter und deren Werten werden unten auf dem Bildschirm aufgelistet. Aufgrund der Anzahl der Parameter müssen Sie möglicherweise nach oben und unten scrollen, um den Parameter zu finden, an dem Sie interessiert sind.

### Auflisten der Werte einer Parametergruppe (AWS CLI)

Um die Parameter einer Parametergruppe und ihre Werte mithilfe von aufzulisten AWS CLI, verwenden Sie den Befehl `describe-cache-parameters`.

#### Example

Der folgende Beispielcode listet alle Memcached-Parameter und ihre Werte für die Parametergruppe `myMem14` auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameters \
 --cache-parameter-group-name myMem14
```

Für Windows:

```
aws elasticache describe-cache-parameters ^
 --cache-parameter-group-name myMem14
```

## Example

Der folgende Beispielcode listet alle Parameter und ihre Werte für die Parametergruppe myRedis 28 auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameters \
 --cache-parameter-group-name myRedis28
```

Für Windows:

```
aws elasticache describe-cache-parameters ^
 --cache-parameter-group-name myRed28
```

Weitere Informationen finden Sie unter [describe-cache-parameters](#).

## Auflisten der Werte einer Parametergruppe (ElastiCache API)

Verwenden Sie die DescribeCacheParameters Aktion, um die Parameter einer Parametergruppe und ihre Werte mithilfe von aufzulisten. ElastiCache API

## Example

Der folgende Beispielcode listet alle Memcached-Parameter für die Parametergruppe myMem 14 auf.

```
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>
```

Die Antwort auf diese Aktion sieht in etwa folgendermaßen aus. Diese Antwort wurde abgeschnitten.

```

<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

Im folgenden Beispielcode sind alle Parameter für die Parametergruppe myRed 28 aufgeführt.

```

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>

```

Die Antwort auf diese Aktion sieht in etwa folgendermaßen aus. Diese Antwort wurde abgeschnitten.

```

<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>
 <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
 </DescribeCacheParametersResponse>

```

...output omitted...

```
</ResponseMetadata>
</DescribeCacheParametersResponse>
```

Weitere Informationen finden Sie unter [DescribeCacheParameters](#).

## Änderung einer ElastiCache Parametergruppe

### Important

Die Standard-Parametergruppe kann nicht abgeändert werden.

Einige Parameterwerte in einer Parametergruppe können bearbeitet werden. Diese Parameterwerte werden auf die zur Parametergruppe gehörenden Cluster angewendet. Weitere Informationen dazu, wann eine Änderung eines Parameterwerts auf eine Parametergruppe angewendet wird, finden Sie unter [Valkey- und Redis-Parameter OSS](#) und [Memcached-spezifische Parameter](#).

### Ändern einer Parametergruppe (Konsole)

Das folgende Verfahren zeigt, wie Sie den Wert des `cluster-enabled` Parameters mithilfe der ElastiCache Konsole ändern können. Dieses Verfahren ist bei der Bearbeitung des Wertes aller Parameter gleich.

So ändern Sie den Wert eines Parameters mithilfe der ElastiCache Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Wählen Sie die Parametergruppe aus, die Sie bearbeiten möchten, indem Sie das Kontrollkästchen links neben dem Namen der Parametergruppe aktivieren.

Die Parameter der Parametergruppe werden unten auf dem Bildschirm aufgelistet. Sie können die Liste seitenweise durchlaufen, um alle Parameter zu sehen.

4. Um einen oder mehrere Parameter zu bearbeiten, wählen Sie Edit Parameters.
5. Scrollen Sie auf dem Bildschirm Edit Parameter Group: mit der linken und rechten Pfeiltaste, bis Sie zum Parameter `binding_protocol` gelangen. Geben Sie dann `ascii` in die Spalte Value ein.

6. Wählen Sie Save Changes.
7. Informationen zum Namen des Parameters, den Sie geändert haben, finden Sie unter [Memcached-spezifische Parameter](#). Wenn als Zeitpunkt für die Übernahme der Parameteränderung After restart festgelegt ist, starten Sie jeden Cluster neu, der von dieser Parametergruppe Gebrauch macht. Weitere Informationen finden Sie unter [Neustarten von Clustern](#).
8. Informationen zum Ermitteln des Namens des Parameters OSS, den Sie bei Valkey und Redis geändert haben, finden Sie unter [Valkey- und Redis-Parameter OSS](#). Wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) haben und Änderungen an den folgenden Parametern vornehmen, müssen Sie die Knoten im Cluster neu starten:
  - activerehashing
  - databases

Weitere Informationen finden Sie unter [Knoten neu starten](#).

 Änderungen der Parameter Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
  - databases
1. Mit Redis können Sie ein manuelles Backup Ihres Clusters erstellen. Siehe [Erstellen manueller Backups](#).
  2. Löschen Sie den Cluster. Siehe [Löschen von Clustern](#).
  3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

## Ändern einer Parametergruppe (AWS CLI)

Um den Wert eines Parameters mit dem zu ändern AWS CLI, verwenden Sie den Befehl. `modify-cache-parameter-group`

### Example

Bei Memcached finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Memcached-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `chunk_size` und `chunk_size_growth_fact` auf der Parametergruppe `myMem14`.

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe dieses Befehls sieht in etwa folgendermaßen aus.

```
{
 "CacheParameterGroupName": "myMem14"
}
```

### Example

Bei Valkey und Redis OSS finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Valkey- und Redis-Parameter OSS](#)

Der folgende Beispielcode legt den Wert von zwei Parametern fest `reserved-memory-percent` und `actived-cluster-mode` für die Parametergruppe `myredis32-on-30`. Wir setzen `reserved-`

memory-percentage auf 30 (30 Prozent) und cluster-enabled auf, yes sodass die Parametergruppe mit Valkey- oder Redis-Clustern OSS (Clustermodus aktiviert) (Replikationsgruppen) verwendet werden kann.

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe dieses Befehls sieht in etwa folgendermaßen aus.

```
{
 "CacheParameterGroupName": "my-redis32-on-30"
}
```

Weitere Informationen finden Sie unter [modify-cache-parameter-group](#).

Den Namen des Parameters, den Sie in einem dieser Themen bearbeitet haben, finden Sie unter [Valkey- und Redis-Parameter OSS](#).

Wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) haben und Änderungen an den folgenden Parametern vornehmen, müssen Sie die Knoten im Cluster neu starten:

- activerehashing
- databases

Weitere Informationen finden Sie unter [Knoten neu starten](#).

### Änderungen der Parameter Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
  - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
  2. Löschen Sie den Cluster. Siehe [Löschen von Clustern](#).
  3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

## Ändern einer Parametergruppe (ElastiCache API)

Verwenden Sie die Aktion, um die Parameterwerte einer Parametergruppe mithilfe von zu ändern.

ElastiCache API ModifyCacheParameterGroup

### Example

Bei Memcached finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Memcached-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `chunk_size` und `chunk_size_growth_fact` auf der Parametergruppe `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

Bei Valkey und Redis OSS finden Sie den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, unter [Valkey- und Redis-Parameter OSS](#)

Der folgende Beispielcode legt den Wert von zwei Parametern fest `reserved-memory-percent` und aktiviert den Cluster-Modus für die Parametergruppe `myredis32-on-30`. Wir setzen `reserved-memory-percent` auf `30` (30 Prozent) und `cluster-enabled` auf `yes` sodass die Parametergruppe mit Valkey- oder Redis-Clustern OSS (Clustermodus aktiviert) (Replikationsgruppen) verwendet werden kann.

```
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>
```

Weitere Informationen finden Sie unter [ModifyCacheParameterGroup](#).

Wenn Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) haben und Änderungen an den folgenden Parametern vornehmen, müssen Sie die Knoten im Cluster neu starten:

- `activeresharding`
- `databases`

Weitere Informationen finden Sie unter [Knoten neu starten](#).

**i** Änderungen der Parameter Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn Sie Änderungen an den folgenden Parametern auf einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) vornehmen, gehen Sie wie folgt vor.

- activerehashing
  - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
  2. Löschen Sie den Cluster. Siehe [Löschen eines Clusters in ElastiCache](#).
  3. Stellen Sie den Cluster mit der abgeänderten Parametergruppe und der Sicherung zum Starten des neuen Clusters wieder her. Siehe [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Bei Änderungen an anderen Parametern ist dies nicht erforderlich.

## Löschen einer ElastiCache Parametergruppe

Sie können eine benutzerdefinierte Parametergruppe mithilfe der ElastiCache Konsole AWS CLI, der oder der löschen ElastiCache API.

Parametergruppen, die Clustern zugeordnet sind, können nicht gelöscht werden. Standard-Parametergruppen können ebenfalls nicht gelöscht werden.

### Löschen einer Parametergruppe (Konsole)

Das folgende Verfahren zeigt, wie Sie eine Parametergruppe mithilfe der ElastiCache Konsole löschen.

Um eine Parametergruppe mit der ElastiCache Konsole zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Wählen Sie die Parametergruppen aus, die Sie löschen möchten, indem Sie das Kontrollkästchen links neben dem Namen der Parametergruppe aktivieren.

Die Schaltfläche Löschen wird aktiv.

4. Wählen Sie Löschen.

Der Bestätigungsbildschirm Delete Parameter Groups wird angezeigt.

5. Zum Löschen der Parametergruppen wählen Sie auf dem Bestätigungsbildschirm Delete Parameter Groups die Option Delete aus.

Um die Parametergruppen beizubehalten, wählen Sie Cancel.

### Löschen einer Parametergruppe (AWS CLI)

Um eine Parametergruppe mit dem zu löschen AWS CLI, verwenden Sie den Befehl `delete-cache-parameter-group`. Der durch `--cache-parameter-group-name` angegebenen Parametergruppe dürfen eine Cluster zugeordnet sein und es darf sich nicht um eine Standard-Parametergruppe handeln, damit sie gelöscht wird.

Der folgende Beispielcode löscht die myMem14-Parametergruppe.

## Example

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-parameter-group \
 --cache-parameter-group-name myRed28
```

Für Windows:

```
aws elasticache delete-cache-parameter-group ^
 --cache-parameter-group-name myRed28
```

Weitere Informationen finden Sie unter [delete-cache-parameter-group](#).

## Löschen einer Parametergruppe (ElastiCache API)

Verwenden Sie die `DeleteCacheParameterGroup` Aktion ElastiCache API, um eine Parametergruppe mithilfe von `CacheParameterGroupName` zu löschen. Der durch `CacheParameterGroupName` angegebenen Parametergruppe dürfen eine Cluster zugeordnet sein und es darf sich nicht um eine Standard-Parametergruppe handeln, damit sie gelöscht wird.

## Example

Bei Memcached löscht der folgende Beispielcode die `myMem14`-Parametergruppe.

```
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

Der folgende Beispielcode löscht die Parametergruppe 28. `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>
```

Weitere Informationen finden Sie unter [DeleteCacheParameterGroup](#).

## Motorspezifische Parameter

### Valkey und Redis OSS

Die Valkey 7.2-Parameter entsprechen den OSS Redis 7-Parametern.

Wenn Sie keine Parametergruppe für Ihren Valkey- oder OSS Redis-Cluster angeben, wird eine Standardparametergruppe verwendet, die Ihrer Engine-Version entspricht. Sie können die Werte von Parametern in der Standard-Parametergruppe nicht ändern. Sie können jedoch jederzeit eine benutzerdefinierte Parametergruppe erstellen und sie Ihrem Cluster zuordnen, solange die Werte von bedingungsabhängig veränderbaren Parametern in beiden Parametergruppen gleich sind. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

### Themen

- [Valkey- und Redis-Parameter OSS](#)
- [Memcached-spezifische Parameter](#)

## Valkey- und Redis-Parameter OSS

### Themen

- [Änderungen der Parameter Valkey 7.2 und OSS Redis 7](#)
- [Änderungen der Parameter in Redis OSS 6.x](#)
- [Änderungen der Parameter in Redis 5.0.3 OSS](#)
- [Änderungen der Parameter in Redis 5.0.0 OSS](#)
- [Änderungen der Parameter in Redis OSS 4.0.10](#)
- [Änderungen der Parameter in Redis OSS 3.2.10](#)
- [Änderungen der Parameter in Redis 3.2.6 OSS](#)
- [Änderungen der Parameter in Redis 3.2.4 OSS](#)
- [Redis OSS 2.8.24 \(erweitert\) hat Parameter hinzugefügt](#)
- [Redis OSS 2.8.23 \(erweitert\) hat Parameter hinzugefügt](#)
- [Redis OSS 2.8.22 \(erweitert\) hat Parameter hinzugefügt](#)
- [Redis 2.8.21 OSS hat Parameter hinzugefügt](#)
- [Redis 2.8.19 OSS hat Parameter hinzugefügt](#)
- [Redis 2.8.6 OSS hat Parameter hinzugefügt](#)
- [Redis 2.6.13-Parameter OSS](#)
- [OSSRedis-Knotentyp-spezifische Parameter](#)

### Änderungen der Parameter Valkey 7.2 und OSS Redis 7

Familie der Parametergruppe: redis7

Die Standardparametergruppen von Redis OSS 7 lauten wie folgt:

- `default.redis7`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis7.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) und Replikationsgruppen.

Die in Redis OSS 7 hinzugefügten Parameter lauten wie folgt.

Name	Details	Beschreibung
<code>cluster-allow-pubsubshard-when-down</code>	<p>Gültige Werte: yes, no</p> <p>Standard: yes</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Wenn die Standardeinstellung „Ja“ lautet, können Knoten Pubsub-Shard-Datenverkehr bereitstellen, während sich der Cluster in einem ausgefallenen Zustand befindet, sofern er glauben, dass er die Slots besitzt.</p>
<code>cluster-preferred-endpoint-type</code>	<p>Gültige Werte: ip, tls-dynamic</p> <p>Standard: tls-dynamic</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Dieser Wert steuert, welcher Endpunkt für MOVED ASKING /-Anfragen zurückgegeben wird, sowie das Endpunktfeld für CLUSTER SLOTS und CLUSTER SHARDS. Wenn der Wert auf „ip“ festgelegt ist, gibt der Knoten seine IP-Adresse bekannt. Wenn der Wert auf tls-dynamic gesetzt ist, gibt der Knoten einen Hostnamen bekannt, wenn er aktiviert encryption-in-transit ist, und andernfalls eine IP-Adresse.</p>
<code>latency-tracking</code>	<p>Gültige Werte: yes, no</p> <p>Standard: no</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Wenn diese Option auf „Ja“ festgelegt ist, werden die Latenzen pro Befehl protokolliert und die Perzentilverteilung über den Latenzstatistiken-Befehl INFO wird aktiviert. Ebenso werden die kumulativen Latenzverteilungen (Histogramme) über den LATENCY-Befehl exportiert.</p>

Name	Details	Beschreibung
hash-max-listpack-entries	<p>Zulässige Werte: 0+</p> <p>Standard: 512</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Die maximale Anzahl von Hash-Einträgen, damit der Datensatz komprimiert werden kann.</p>
hash-max-listpack-value	<p>Zulässige Werte: 0+</p> <p>Standard: 64</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Der Schwellenwert der maximalen Anzahl von Hash-Einträgen, damit der Datensatz komprimiert werden kann.</p>
zset-max-listpack-entries	<p>Zulässige Werte: 0+</p> <p>Standard: 128</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Die maximale Anzahl von Einträgen zu sortierten Sätzen, damit der Datensatz komprimiert werden kann.</p>

Name	Details	Beschreibung
zset-max-listpack-value	<p>Zulässige Werte: 0+</p> <p>Standard: 64</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	Der Schwellenwert der maximalen Anzahl von Einträgen zu sortierten Sätzen, damit der Datensatz komprimiert werden kann.

Die in Redis OSS 7 geänderten Parameter lauten wie folgt.

Name	Details	Beschreibung
activerehashing	<p>Anpassbar: no. In Redis OSS 7 ist dieser Parameter standardmäßig ausgeblendet und aktiviert. Wenn Sie ihn deaktivieren möchten, müssen Sie einen <a href="#">Support-Fall erstellen</a>.</p>	Veränderbar war ja.

In Redis OSS 7 wurden die folgenden Parameter entfernt.

Name	Details	Beschreibung
hash-max-ziplist-entries	<p>Zulässige Werte: 0+</p> <p>Standard: 512</p> <p>Typ: Ganzzahl</p>	Für die Darstellung kleiner Hash-Kodierungen listpack statt ziplist verwenden

Name	Details	Beschreibung
	<p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	
<p>hash-max-ziplist-value</p>	<p>Zulässige Werte: 0+</p> <p>Standard: 64</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Für die Darstellung kleiner Hash-Kodierungen listpack statt ziplist verwenden</p>
<p>zset-max-ziplist-entries</p>	<p>Zulässige Werte: 0+</p> <p>Standard: 128</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Verwenden Sie listpack statt ziplist für die Darstellung kleiner Hash-Kodierungen.</p>

Name	Details	Beschreibung
<code>zset-max-ziplist-value</code>	<p>Zulässige Werte: 0+</p> <p>Standard: 64</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	Verwenden Sie <code>listpack</code> statt <code>ziplist</code> für die Darstellung kleiner Hash-Kodierungen.
<code>list-max-ziplist-size</code>	<p>Gültige Werte:</p> <p>Standard: -2</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	Die Anzahl der zulässigen Einträge pro internem Listenknoten.

## Änderungen der Parameter in Redis OSS 6.x

Familie der Parametergruppe: `redis6.x`

Die Standardparametergruppen von Redis OSS 6.x lauten wie folgt:

- `default.redis6.x`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis6.x.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) und Replikationsgruppen.

**Note**

In der OSS Redis-Engine-Version 6.2, als die R6GD-Node-Familie für die Verwendung mit eingeführt wurde, wurden nur die Max-Speicher-Richtlinien Noeviction [Daten-Tiering ElastiCache](#), Volatile-Lru und Allkeys-LRU für r6gd-Knotentypen unterstützt.

Weitere Informationen erhalten Sie unter [ElastiCache \(RedisOSS\) Version 6.2 \(erweitert\)](#) und [ElastiCache \(RedisOSS\) Version 6.0 \(erweitert\)](#).

Die in Redis OSS 6.x hinzugefügten Parameter lauten wie folgt.

Details	Beschreibung	
<code>acl-pubsub-default</code> (added in 6.2)	<p>Gültige Werte: <code>resetchannels</code> , <code>allchannels</code></p> <p>Standard: <code>allchannels</code></p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Die Änderungen werden wirksam: Die vorhandenen OSS Redis-Benutzer, die dem Cluster zugeordnet sind, werden weiterhin über bestehende Berechtigungen verfügen. Aktualisieren Sie entweder die Benutzer oder starten Sie den Cluster neu, um die vorhandenen OSS Redis-Benutzer zu aktualisieren.</p>	Standard-Pubsub-Kanalberechtigungen für ACL Benutzer, die in diesem Cluster bereitgestellt sind.
<code>cluster-allow-read</code>	Standard: <code>no</code>	Wenn diese Option auf Ja gesetzt ist, verarbeitet eine Redis-Replikationsgruppe OSS

Details	Beschreibung	
<p>s-when-down (added in 6.0)</p>	<p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>(Clustermodus aktiviert) weiterhin Lesebefehle, auch wenn ein Knoten nicht in der Lage ist, ein Quorum von Primärdaten zu erreichen.</p> <p>Wenn der Standardwert Nein festgelegt ist, lehnt die Replikationsgruppe alle Befehle ab. Wir empfehlen, diesen Wert auf yes zu setzen, wenn Sie einen Cluster mit weniger als drei Knotengruppen verwenden oder Ihre Anwendung veraltete Lesevorgänge sicher verarbeiten kann.</p>
<p>tracking-table-max-keys (added in 6.0)</p>	<p>Standard: 1.000.000</p> <p>Typ: Zahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Um das clientseitige Caching zu unterstützen, OSS unterstützt Redis die Nachverfolgung, welche Clients auf welche Schlüssel zugegriffen haben.</p> <p>Wenn der verfolgte Schlüssel geändert wird, werden Invalidierungsnachrichten an alle Clients gesendet, um ihnen mitzuteilen, dass ihre zwischengespeicherten Werte nicht mehr gültig sind. Mit diesem Wert können Sie die Obergrenze dieser Tabelle angeben. Nachdem dieser Parameterwert überschritten wurde, werden Clients zufällig Invalidierung gesendet. Dieser Wert sollte so eingestellt werden, dass die Speicherauslastung begrenzt wird, während immer noch genügend Schlüssel verfolgt werden. Schlüssel werden auch unter niedrigen Speicherbedingungen ungültig gemacht.</p>

Details	Beschreibung	
<p><code>aclog-max-len</code> (added in 6.0)</p>	<p>Standard: 128</p> <p>Typ: Zahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Dieser Wert entspricht der maximalen Anzahl von Einträgen im Protokoll. ACL</p>
<p><code>active-expire-effort</code> (added in 6.0)</p>	<p>Standard: 1</p> <p>Typ: Zahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Redis OSS löscht Schlüssel, deren Gültigkeitsdauer überschritten wurde, auf zwei Arten. In einem wird auf einen Schlüssel zugegriffen und festgestellt, dass er abgelaufen ist. In der anderen, ein periodischer Job Proben Schlüssel und bewirkt, dass diejenigen, die ihre Zeit überschritten haben, ablaufen. Dieser Parameter definiert den Aufwand, den Redis OSS aufwendet, um Elemente im periodischen Job ablaufen zu lassen.</p> <p>Der Standardwert von 1 versucht zu vermeiden, dass mehr als 10 Prozent der abgelaufenen Schlüssel noch im Speicher sind. Außerdem wird versucht, mehr als 25 Prozent des gesamten Arbeitsspeichers zu verbrauchen und das System Latenz zu erhöhen. Sie können diesen Wert auf bis zu 10 erhöhen, um den Aufwand für ablaufende Schlüssel zu erhöhen. Der Kompromiss besteht in einer höheren CPU und potenziell höheren Latenz. Wir empfehlen einen Wert von 1, es sei denn, Sie stellen eine hohe Speicherauslastung fest und können eine Erhöhung CPU der Auslastung tolerieren.</p>

Details	Beschreibung	
lazyfree-lazy-user-del (added in 6.0)	Standard: no  Typ: Zeichenfolge  Veränderbar: Ja  Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster	Wenn der Wert auf Ja gesetzt ist, verhält sich der DEL-Befehl genauso wie UNLINK.

In Redis OSS 6.x wurden die folgenden Parameter entfernt.

Name	Details	Beschreibung
lua-replique-commands	Zulässige Werte: yes/no  Standard: yes  Typ: Boolesch  Veränderbar: Ja  Änderungen werden wirksam: Sofort	Diese Parameter aktivieren immer Lua-Effektreplikation oder Nicht-in-Lua-Skripts.

Änderungen der Parameter in Redis 5.0.3 OSS

Familie der Parametergruppe: redis5.0

Redis OSS 5.0-Standardparametergruppen

- `default.redis5.0`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) und Replikationsgruppen.

- `default.redis5.0.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) und Replikationsgruppen.

### In Redis 5.0.3 hinzugefügte Parameter OSS

Name	Details	Beschreibung
<code>rename-commands</code>	<p>Standard: keiner</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p>	<p>Eine durch Leerzeichen getrennte Liste umbenannter Redis-Befehle. OSS Im Folgenden finden Sie eine eingeschränkte Liste der verfügbaren Befehle:</p> <p>APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BRPOPLPUSH BZPOPMIN BZPOPMAX CLIENT CLUSTER COMMAND DBSIZE DECR 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 HINCRBYFLOAT HKEYS HLEN HMGET HMSET HSET HSETNX HSTRLEN HVALS INCR INCRBY INCRBYFLOAT INFO KEYS LASTSAVE LINDEX LINSERT LLEN LPOP LPUSSH 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</p>

Name	Details	Beschreibung
		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 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

Weitere Informationen finden Sie unter [ElastiCache \(RedisOSS\) Version 5.0.6 \(erweitert\)](#).

## Änderungen der Parameter in Redis 5.0.0 OSS

Familie der Parametergruppe: redis5.0

### Redis OSS 5.0-Standardparametergruppen

- `default.redis5.0`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) und Replikationsgruppen.
- `default.redis5.0.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) und Replikationsgruppen.

## In Redis 5.0 hinzugefügte Parameter OSS

Name	Details	Beschreibung
<code>stream-node-max-bytes</code>	<p>Zulässige Werte: 0+</p> <p>Standard: 4096</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Struktur der Stream-Daten ist eine baumartige Struktur von Knoten, die mehrere Elemente im Innern codieren. Mit dieser Konfiguration legen Sie die maximale Größe (in Bytes) eines einzelnen Knotens in der Baumstruktur fest. Wird die Einstellung „0“ gewählt, ist die Größe des Baumknotens unbegrenzt.</p>
<code>stream-node-max-entries</code>	<p>Zulässige Werte: 0+</p> <p>Standard: 100</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Struktur der Stream-Daten ist eine baumartige Struktur von Knoten, die mehrere Elemente im Innern codieren. Legen Sie mit dieser Konfiguration die maximale Anzahl der Elemente fest, die ein einzelner Knoten enthalten kann, bevor beim Anhängen neuer Stream-Einträge zu einem neuen Knoten gewechselt wird. Wird die Einstellung „0“ gewählt, ist die Anzahl der Elemente des Baumknotens unbegrenzt.</p>
<code>active-defrag-max-scan-fields</code>	<p>Zulässige Werte: 1 bis 1 000 000</p> <p>Standard: 1000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Maximale Anzahl an Set-/Hash-/Zset-/Listenfeldern, die vom Scan des Hauptwörterbuchs verarbeitet werden</p>

Name	Details	Beschreibung
<code>lua-repl cate-comm ands</code>	Zulässige Werte: yes/no  Standard: yes  Typ: Boolesch  Veränderbar: Ja  Änderungen werden wirksam: Sofort	Diese Parameter aktivieren immer Lua-Effektreplikation oder Nicht-in-Lua-Skripts.
<code>replica-i gnore-max memory</code>	Standard: yes  Typ: Boolesch  Veränderbar: Nein	Dieser Parameter bestimmt, ob das Replikat die Einstellung <code>maxmemory</code> ignoriert, indem Elemente nicht unabhängig von der Primär-Instance bereinigt werden

Redis OSS hat als Reaktion auf das Feedback der Community mehrere Parameter in Engine-Version 5.0 umbenannt. Weitere Informationen finden Sie unter [Was ist neu in Redis 5 OSS?](#) . In der folgenden Tabelle sind die neuen Parameternamen und deren Entsprechung in früheren Versionen angegeben.

Parameter wurden in Redis 5.0 OSS umbenannt

Name	Details	Beschreibung
<code>replica-l azy-flush</code>	Standard: yes  Typ: Boolesch  Veränderbar: Nein  Früherer Name: <code>slave-lazy-flush</code>	Führt einen asynchronen <code>flushDB</code> -Befehl während einer Replikatsynchronisierung durch

Name	Details	Beschreibung
<code>client-output-buffer-limit-replica-hard-limit</code>	<p>Standard: Für Werte siehe <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a></p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>Früherer Name: <code>client-output-buffer-limit - slave-hard-limit</code></p>	Für OSS Redis-Read-Repliken: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird der Client getrennt.
<code>client-output-buffer-limit-replica-soft-limit</code>	<p>Standard: Für Werte siehe <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a></p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>Früherer Name: - <code>client-output-buffer-limit slave-soft-limit</code></p>	Für OSS Redis-Read-Replicas: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird die Verbindung zum Client unterbrochen, aber nur, wenn dieser Zustand weiterhin besteht. <code>client-output-buffer-limit-replica-soft-seconds</code>
<code>client-output-buffer-limit-replica-soft-seconds</code>	<p>Standard: 60</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>Früherer Name: - <code>client-output-buffer-limit slave-soft-seconds</code></p>	Für OSS Redis-Read-Replicas: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden auf <code>client-output-buffer-limit-replica-soft-limit</code> Byte verbleibt, wird die Verbindung zum Client unterbrochen.

Name	Details	Beschreibung
<code>replica-allow-chaining</code>	<p>Standard: no</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Nein</p> <p>Früherer Name: slave-allow-chaining</p>	<p>Bestimmt, ob eine Read Replica in Redis eigene Read Replicas haben OSS kann.</p>
<code>min-replicas-to-write</code>	<p>Standard: 0</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Früherer Name: min-slaves-to-write</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Mindestanzahl an Read Replicas, die verfügbar sein müssen, damit der primäre Knoten Schreibanforderungen von Clients akzeptiert. Wenn die Anzahl der verfügbaren Replikate unter diesen Wert abfällt, dann akzeptiert der primäre Knoten keine Schreibanforderungen mehr.</p> <p>Wenn entweder dieser Parameter oder 0 <code>min-replicas-max-lag</code> ist, akzeptiert der primäre Knoten immer Schreibanforderungen, auch wenn keine Replikate verfügbar sind.</p>

Name	Details	Beschreibung
<code>min-replicas-max-lag</code>	<p>Standard: 10</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Früherer Name: <code>min-slaves-max-lag</code></p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Anzahl von Sekunden, innerhalb der der primäre Knoten eine Ping-Anforderung von einer Read Replica empfangen muss. Wenn dieser Zeitraum überschritten wird, ohne dass der primäre Knoten einen Ping erhält, dann wird das Replikat nicht mehr als verfügbar angesehen. Wenn die Anzahl der verfügbaren Replikate unter den Wert fällt <code>min-replicas-to-write</code>, akzeptiert das primäre Replikat zu diesem Zeitpunkt keine Schreibvorgänge mehr.</p> <p>Wenn entweder dieser Parameter oder 0 <code>min-replicas-to-write</code> ist, akzeptiert der primäre Knoten immer Schreibanforderungen, auch wenn keine Replikate verfügbar sind.</p>
<code>close-on-replica-write</code>	<p>Standard: yes</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Früherer Name: <code>close-on-slave-write</code></p> <p>Änderungen werden wirksam: Sofort</p>	<p>Wenn er aktiviert ist, wird die Verbindung mit Clients, die versuchen, in ein schreibgeschütztes Replikat zu schreiben, unterbrochen.</p>

### In Redis OSS 5.0 entfernte Parameter

Name	Details	Beschreibung
<code>repl-timeout</code>	<p>Standard: 60</p> <p>Veränderbar: Nein</p>	<p>Parameter ist in dieser Version nicht verfügbar</p>

## Änderungen der Parameter in Redis OSS 4.0.10

Familie der Parametergruppe: redis4.0

Standardparametergruppen von Redis 4.0.x OSS

- `default.redis4.0`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) und Replikationsgruppen.
- `default.redis4.0.cluster.on`— Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) und Replikationsgruppen.

Die Parameter wurden in Redis 4.0.10 geändert OSS

Name	Details	Beschreibung
<code>maxmemory-policy</code>	<p>Gültige Werte: <code>allkeys-lru</code> , <code>volatile-lru</code> , <b><code>allkeys-lfu</code></b> , <b><code>volatile-lfu</code></b> , <code>allkeys-random</code> , <code>volatile-random</code> , <code>volatile-ttl</code> , <code>noeviction</code></p> <p>Standard: <code>volatile-lru</code></p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: sofort</p>	<p><code>maxmemory-policy</code> wurde in Version 2.6.13 hinzugefügt. In Version 4.0.10 wurden zwei neue zulässige Werte hinzugefügt: <code>allkeys-lfu</code> , wodurch alle Schlüssel mit <code>approximated</code> entfernt werden, und <code>LFUvolatile-lfu</code> , die unter Verwendung von <code>approximated</code> unter den Schlüsseln mit einem <code>Expire-Set</code> entfernt werden. <code>LFU</code> In Version 6.2, als die <code>R6gd-Knot</code> eingeführt wurde, werden nur die <code>maxmemory-Richtlinien</code> <code>noeviction</code> , <code>volatile-lru</code> und <code>allkeys-lru</code> mit <code>R6gd-Knotentypen</code> unterstützt.</p>

In OSS Redis 4.0.10 wurden Parameter hinzugefügt

Name	Details	Beschreibung
Parameter für asynchrones Löschen		

Name	Details	Beschreibung
lazyfree-lazy- eviction	Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei Bereinigungen durch.
lazyfree-lazy-expire	Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei abgelaufenen Schlüsseln durch.
lazyfree-lazy-server-del	Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort	Führt eine asynchrone Löschung bei Befehlen durch, die Werte aktualisieren.

Name	Details	Beschreibung
<code>slave-lazy-flush</code>	<p>Zulässige Werte: Nicht zutreffend</p> <p>Standard: no</p> <p>Typ: Boolesch</p> <p>Veränderbar: Nein</p> <p>Änderungen treten in Kraft: Nicht zutreffend</p>	<p>Führt einen asynchrones flushDB-Befehl während einer Slave-Synchronisierung durch.</p>
LFUParameter		
<code>lfu-log-factor</code>	<p>Zulässige Werte: jede Ganzzahl</p> <p>Standard: 10</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: sofort</p>	<p>Legen den Protokollfaktor fest, der die Anzahl der Schlüssel treffer zur Deckung des Schlüsselzählers bestimmt.</p>
<code>lfu-decay-time</code>	<p>Zulässige Werte: jede Ganzzahl</p> <p>Standard: 1</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: sofort</p>	<p>Zeitdauer in Minuten für die Verringerung des Schlüssel zählers.</p>

Name	Details	Beschreibung
Parameter für aktive Defragmentierung		
<code>activedefrag</code>	Zulässige Werte: yes/no  Standard: no  Typ: Boolesch  Veränderbar: Ja  Änderungen treten in Kraft: sofort	Aktivierte aktive Defragmentierung
<code>active-defrag-ignore-bytes</code>	Zulässige Werte: 10485760-104857600  Standard: 104857600  Typ: Ganzzahl  Veränderbar: Ja  Änderungen treten in Kraft: sofort	Mindestmenge an Fragmentierungsresten für das Starten der aktiven Defragmentierung.
<code>active-defrag-threshold-lower</code>	Zulässige Werte: 1–100  Standard: 10  Typ: Ganzzahl  Veränderbar: Ja  Änderungen treten in Kraft: sofort	Mindestprozentsatz der Fragmentierung zum Starten der aktiven Defragmentierung.

Name	Details	Beschreibung
<code>active-defrag-threshold-upper</code>	Zulässige Werte: 1–100 Standard: 100 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Maximaler Prozentsatz der Fragmentierung, bei dem der maximale Einsatz aufgewandt wird.
<code>active-defrag-cycle-min</code>	Zulässige Werte: 1–75 Standard: 25 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Minimaler Aufwand für die Defragmentierung in CPU Prozent.
<code>active-defrag-cycle-max</code>	Zulässige Werte: 1–75 Standard: 75 Typ: Ganzzahl Veränderbar: Ja Änderungen treten in Kraft: sofort	Maximaler Aufwand für die Defragmentierung in CPU Prozent.

#### Client-Ausgabepuffer-Parameter

Name	Details	Beschreibung
<code>client-query-buffer-limit</code>	<p>Zulässige Werte: 1048576-1073741824</p> <p>Standard: 1073741824</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: sofort</p>	Maximale Größe eines einzelnen Client-Abfragepuffers.
<code>proto-max-bulk-len</code>	<p>Zulässige Werte: 1048576-536870912</p> <p>Standard: 536870912</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: sofort</p>	Maximale Größe einer einzelnen Elementanforderung.

#### Änderungen der Parameter in Redis OSS 3.2.10

Familie der Parametergruppe: redis3.2

ElastiCache (RedisOSS) 3.2.10 Es werden keine zusätzlichen Parameter unterstützt.

#### Änderungen der Parameter in Redis 3.2.6 OSS

Familie der Parametergruppe: redis3.2

Für Redis OSS 3.2.6 werden keine zusätzlichen Parameter unterstützt.

#### Änderungen der Parameter in Redis 3.2.4 OSS

Familie der Parametergruppe: redis3.2

Ab Redis OSS 3.2.4 gibt es zwei Standardparametergruppen.

- `default.redis3.2`— Wenn Sie Redis OSS 3.2.4 ausführen, geben Sie diese oder eine davon abgeleitete Parametergruppe an, wenn Sie eine Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus deaktiviert) erstellen und trotzdem die zusätzlichen Funktionen von Redis 3.2.4 verwenden möchten. OSS
- `default.redis3.2.cluster.on`— Geben Sie diese oder eine davon abgeleitete Parametergruppe an, wenn Sie eine Valkey- oder OSS Redis-Replikationsgruppe (Clustermodus aktiviert) erstellen möchten.

## Themen

- [Neue Parameter für Redis 3.2.4 OSS](#)
- [Die Parameter wurden in Redis OSS 3.2.4 geändert \(verbessert\)](#)

## Neue Parameter für Redis 3.2.4 OSS

Familie der Parametergruppe: `redis3.2`

Für Redis OSS 3.2.4 werden die folgenden zusätzlichen Parameter unterstützt.

Name	Details	Beschreibung
<code>list-max-ziplist-size</code>	Standard: -2 Typ: Ganzzahl Veränderbar: Nein	Listen sind auf besondere Weise codiert, um Platz zu sparen. Die Anzahl der Einträge, die pro internem Listenknoten zulässig sind, können als eine feste maximale Größe oder als maximale Anzahl von Elementen angegeben werden. Verwenden Sie für eine feste maximale Größe -5 bis -1 mit der folgenden Bedeutung: <ul style="list-style-type: none"> <li>• -5: max. Größe: 64 Kb – wird nicht für normale Arbeitslasten empfohlen</li> <li>• -4: max. Größe: 32 Kb – wird nicht empfohlen</li> <li>•</li> </ul>

Name	Details	Beschreibung
		<p>-3: max. Größe: 16 Kb – wird nicht empfohlen</p> <ul style="list-style-type: none"><li>• -2: max. Größe: 8 Kb – wird empfohlen</li><li>• -1: max. Größe: 4 Kb – wird empfohlen</li><li>• Positive Zahlen bedeuten, bis hin zu genau dieser Anzahl von Elementen pro Listenknoten zu speichern.</li></ul>

Name	Details	Beschreibung
list-compress-depth	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	<p>Listen können auch komprimiert werden. Die Komprimierungstiefe ist die Anzahl der quicklist ziplist-Knoten ab jeder Seite der Liste, die von der Komprimierung ausgeschlossen werden sollen. Anfang und Ende der Liste sind für schnelle push- und pop-Operationen immer dekomprimiert. Die Einstellungen sind:</p> <ul style="list-style-type: none"> <li>• 0: Gesamte Komprimierung deaktivieren.</li> <li>• 1: Komprimierung jeweils einen Knoten nach innen vom Anfang und Ende starten.  <pre>[head]-&gt;node-&gt;node-&gt;...-&gt;node-&gt;[tail]</pre>           Alle Knoten außer [head] und [tail] werden komprimiert.</li> <li>• 2: Komprimierung jeweils zwei Knoten nach innen vom Anfang und Ende starten.  <pre>[head]-&gt;[next]-&gt;node-&gt;node-&gt;...-&gt;node-&gt;[prev]-&gt;[tail]</pre>           [head], [next], [prev], [tail] werden nicht komprimiert. Alle anderen Knoten werden komprimiert.</li> <li>• usw.</li> </ul>

Name	Details	Beschreibung
<code>cluster-enabled</code>	Standard: no/yes * Typ: Zeichenfolge Veränderbar: Nein	<p>Gibt an, ob es sich um eine Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus aktiviert) im Cluster-Modus (ja) oder um eine Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus aktiviert) im Nicht-Cluster-Modus (nein) handelt. Valkey- oder Redis-Replikationsgruppen OSS (Clustermodus aktiviert) im Clustermodus können ihre Daten auf bis zu 500 Knotengruppen partitionieren.</p> <p>* Redis 3.2.OSS. x hat zwei Standardparametergruppen.</p> <ul style="list-style-type: none"> <li>• <code>default.redis3.2</code> – Standardwert no.</li> <li>• <code>default.redis3.2.cluster.on</code> – Standardwert yes.</li> </ul> <p>.</p>

Name	Details	Beschreibung
<code>cluster-require-full-coverage</code>	Standard: no Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Sofort	<p>Wenn diese Option auf <code>yes</code> gesetzt ist, nehmen Valkey- oder Redis-Knoten OSS (Cluster-Modus aktiviert) im Cluster-Modus keine Anfragen mehr an, wenn sie feststellen, dass mindestens ein Hash-Slot aufgedeckt ist (kein verfügbarer Knoten bedient ihn). Auf diese Weise ist ein Cluster nicht mehr verfügbar, falls er teilweise ausfällt. Er wird automatisch wieder verfügbar, sobald alle Slots wieder abgedeckt sind.</p> <p>Manchmal ist es jedoch wünschenswert, dass der Teilbereich des Clusters, der funktioniert, für den immer noch abgedeckten Teil des Keyspace weiterhin Abfragen akzeptiert. Stellen Sie dazu die Option <code>cluster-require-full-coverage</code> auf <code>no</code> ein.</p>

Name	Details	Beschreibung
hll-sparse-max-bytes	<p>Standard: 3000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>HyperLogLog Limit für spärliche Repräsentations-Bytes. Das Limit umfasst den 16-Byte-Header. Wenn eine Darstellung HyperLogLog mit geringer Dichte diese Grenze überschreitet, wird sie in eine Darstellung mit hoher Dichte umgewandelt.</p> <p>Ein Wert größer als 16000 wird nicht empfohlen, da ab einem solchen Wert die dichte Repräsentation effizienter ist.</p> <p>Wir empfehlen einen Wert von etwa 3000, um die Vorteile der platzsparenden Kodierung nutzen zu können, ohne dabei PFADD zu stark zu verlangsamen, was bei der Sparse-Kodierung <math>O(N)</math> ist. Der Wert kann auf <math>\sim 10000</math> erhöht werden, wenn CPU dies kein Problem darstellt, der Leerraum jedoch schon, und der Datensatz besteht aus vielen Datensätzen HyperLogLogs mit einer Kardinalität im Bereich von 0 bis 15000.</p>

Name	Details	Beschreibung
reserved-memory-percent	<p>Standard: 25</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Der reservierte Prozentsatz des Knotenspeichers, der nicht für Daten verwendet werden soll. Standardmäßig wächst der OSS Redis-Datenbedarf, bis er den gesamten Speicher des Knotens verbraucht. In diesem Fall wird die Knotenleistung aufgrund der übermäßigen Speicherauslagerung beeinträchtigt. Indem Sie Speicher reservieren, können Sie einen Teil des verfügbaren Speichers für andere OSS Zwecke als Redis reservieren, um den Umfang des Paging zu reduzieren.</p> <p>Dieser Parameter ist spezifisch für die ElastiCache Redis-Standarddistribution und nicht Teil der Standarddistribution. OSS</p> <p>Weitere Informationen erhalten Sie unter <a href="#">reserved-memory</a> und <a href="#">Verwaltung des reservierten Speichers für Valkey und Redis OSS</a>.</p>

Die Parameter wurden in Redis OSS 3.2.4 geändert (verbessert)

Familie der Parametergruppe: redis3.2

Für Redis OSS 3.2.4 wurden die folgenden Parameter geändert.

Name	Details	Änderung
activeresharding	<p>Veränderbar: Ja, wenn die Parametergruppe keinem Cache-Cluster zugeordnet ist. Ansonsten nein.</p>	<p>Veränderbar war Nein.</p>

Name	Details	Änderung
databases	Veränderbar: Ja, wenn die Parametergruppe keinem Cache-Cluster zugeordnet ist. Ansonsten nein.	Veränderbar war Nein.
appendonly	Standard: deaktiviert Veränderbar: Nein	Wenn Sie ein Upgrade von einer früheren OSS Redis-Version durchführen möchten, müssen Sie das Gerät zuerst ausschalten. appendonly
appendfsync	Standard: deaktiviert Veränderbar: Nein	Wenn Sie ein Upgrade von einer früheren OSS Redis-Version durchführen möchten, müssen Sie das Gerät zuerst ausschaltenappendfsync .
repl-timeout	Standard: 60 Veränderbar: Nein	Ist nun nicht mehr veränderbar mit einem Standard von 60.
tcp-keepalive	Standard: 300	Standard war 0.
list-max-ziplist-entries		Der Parameter ist nicht mehr verfügbar.
list-max-ziplist-value		Der Parameter ist nicht mehr verfügbar.

Redis OSS 2.8.24 (erweitert) hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.24 werden keine zusätzlichen Parameter unterstützt.

## Redis OSS 2.8.23 (erweitert) hat Parameter hinzugefügt

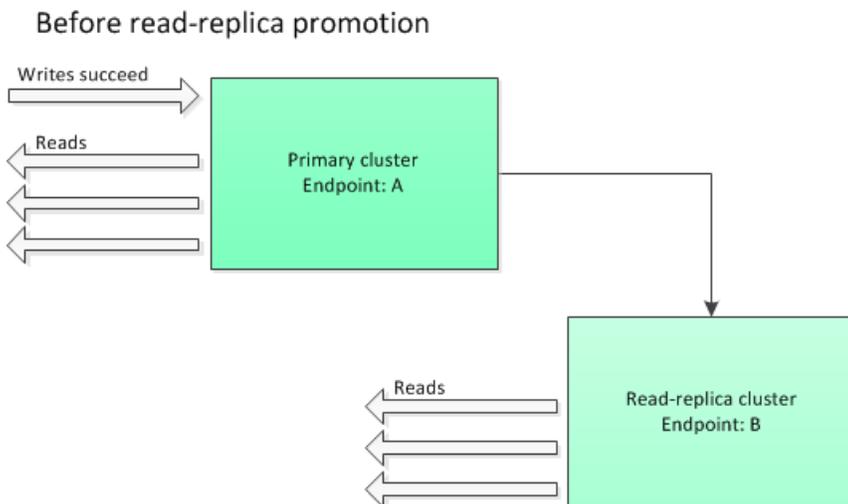
Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.23 wird der folgende zusätzliche Parameter unterstützt.

Name	Details	Beschreibung
<code>close-on-slave-write</code>	Standard: <code>yes</code> Typ: Zeichenfolge ( <code>yes/no</code> ) Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn er aktiviert ist, wird die Verbindung mit Clients, die versuchen, in ein schreibgeschütztes Replikat zu schreiben, unterbrochen.

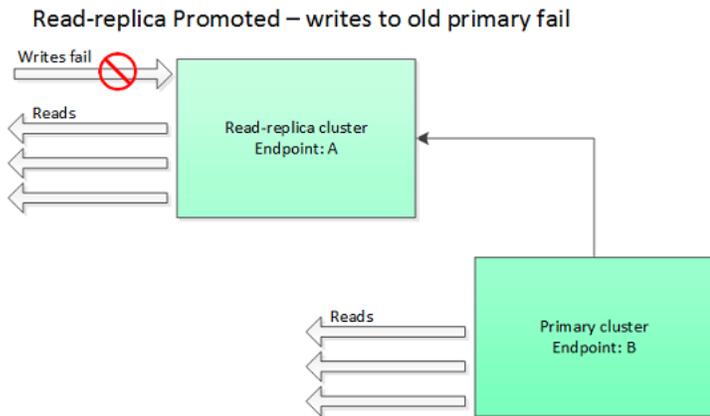
### `close-on-slave-write` Wie funktioniert

Der `close-on-slave-write` Parameter wurde von Amazon eingeführt, ElastiCache um Ihnen mehr Kontrolle darüber zu geben, wie Ihr Cluster reagiert, wenn ein primärer Knoten und ein Read Replica-Knoten aufgrund der Heraufstufung einer Read Replica zum primären Knoten die Rollen wechseln.

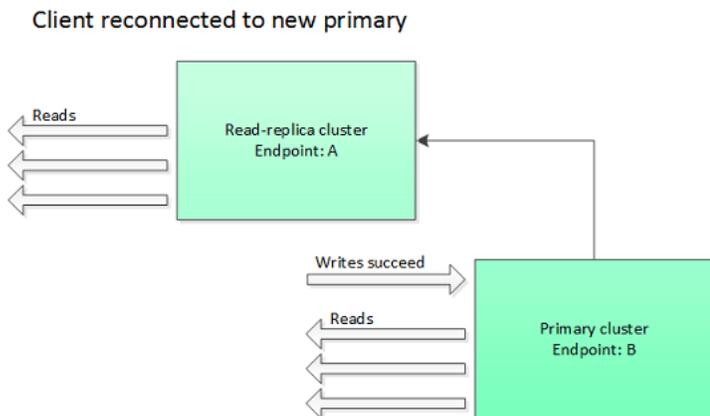


Wenn der Read Replica-Cluster nur aufgrund des Failovers einer Multi-AZ-fähigen Replikation heraufgestuft wird, versucht der Client weiterhin, in den Endpunkt A zu schreiben. Da Endpunkt A nun der Endpunkt für eine Read Replica ist, schlagen diese Schreibvorgänge fehl. Dies ist das Verhalten

für Redis OSS vor der ElastiCache Einführung `close-on-replica-write` und das Verhalten, wenn Sie es deaktivieren `close-on-replica-write`.



Wenn `close-on-replica-write` aktiviert ist, wird jedes Mal bei dem Versuch des Clients, in eine Read Replica zu schreiben, die Client-Verbindung mit dem Cluster unterbrochen. Ihre Anwendungslogik sollte die Verbindungsunterbrechung erkennen, die DNS Tabelle überprüfen und die Verbindung zum primären Endpunkt, der jetzt Endpunkt B wäre, wieder herstellen.



Wann könnten Sie deaktivieren `close-on-replica-write`

Wenn `close-on-replica-write`-Resultate beim Schreiben in fehlerhafte Cluster deaktiviert werden, warum sollte `close-on-replica-write` deaktiviert werden?

Wie zuvor erwähnt, wird jedes Mal, wenn `close-on-replica-write` aktiviert ist, bei dem Versuch des Clients, in eine Read Replica zu schreiben, die Client-Verbindung mit dem Cluster unterbrochen. Das Herstellen einer neuen Verbindung zum Knoten nimmt einige Zeit in Anspruch. Daher beeinträchtigt das Trennen und Wiederherstellen einer Verbindung als Ergebnis einer Schreibanforderung in das Replica auch die Latenz von Leseanforderungen, die über die gleiche Verbindung bedient werden. Dieser Effekt bleibt bestehen bis eine neue Verbindung hergestellt wird.

Wenn Ihre Anwendung besonders leseintensiv oder latenzempfindlich ist, können Sie die Verbindung Ihrer Clients aufrecht erhalten, um eine Beeinträchtigung der Leseleistung zu vermeiden.

Redis OSS 2.8.22 (erweitert) hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.22 werden keine zusätzlichen Parameter unterstützt.

 **Important**

- `repl-backlog-size` Gilt ab OSS Redis-Version 2.8.22 sowohl für den primären Cluster als auch für Replikatcluster.
- Ab OSS Redis-Version 2.8.22 wird der Parameter nicht unterstützt. `repl-timeout` Wenn er geändert wird, ElastiCache wird er mit der Standardeinstellung (60s) überschrieben, wie wir es bei tun. `appendonly`

Die folgenden Parameter werden nicht mehr unterstützt.

- `appendonly`
- `appendfsync`
- `repl-timeout`

Redis 2.8.21 OSS hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.21 werden keine zusätzlichen Parameter unterstützt.

Redis 2.8.19 OSS hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.19 werden keine zusätzlichen Parameter unterstützt.

Redis 2.8.6 OSS hat Parameter hinzugefügt

Familie der Parametergruppe: redis2.8

Für Redis OSS 2.8.6 werden die folgenden zusätzlichen Parameter unterstützt.

Name	Details	Beschreibung
<code>min-slaves-max-lag</code>	<p>Standard: 10</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Anzahl von Sekunden, innerhalb der der primäre Knoten eine Ping-Anforderung von einer Read Replica empfangen muss. Wenn dieser Zeitraum überschritten wird, ohne dass der primäre Knoten einen Ping erhält, dann wird das Replikat nicht mehr als verfügbar angesehen. Wenn die Anzahl der verfügbaren Replikate unter den Wert fällt <code>min-slaves-to-write</code>, akzeptiert das primäre Replikat zu diesem Zeitpunkt keine Schreibvorgänge mehr.</p> <p>Wenn entweder dieser Parameter oder 0 <code>min-slaves-to-write</code> ist, akzeptiert der primäre Knoten immer Schreibanforderungen, auch wenn keine Replikate verfügbar sind.</p>
<code>min-slaves-to-write</code>	<p>Standard: 0</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Mindestanzahl an Read Replicas, die verfügbar sein müssen, damit der primäre Knoten Schreibanforderungen von Clients akzeptiert. Wenn die Anzahl der verfügbaren Replikate unter diesen Wert abfällt, dann akzeptiert der primäre Knoten keine Schreibanforderungen mehr.</p> <p>Wenn entweder dieser Parameter oder 0 <code>min-slaves-max-lag</code> ist, akzeptiert der primäre Knoten</p>

Name	Details	Beschreibung
		immer Schreibenanforderungen, auch wenn keine Replikate verfügbar sind.

Name	Details	Beschreibung
notify-keyspace-events	<p>Standardwert: (eine leere Zeichenfolge)</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Arten von Schlüsselraumereignissen, über die Redis Clients benachrichtigen OSS kann. Jede Art von Ereignis wird durch einen einzelnen Buchstaben repräsentiert:</p> <ul style="list-style-type: none"> <li>• K— Keyspace-Ereignisse, die mit dem Präfix __keyspace@&lt;db&gt; __</li> <li>• E – Key-event-Ereignisse, die mit dem Präfix __keyevent@&lt;db&gt;__ veröffentlicht werden</li> <li>• g — Generische, unspezifische Befehle wie DEL, EXPIRE, usw. RENAME</li> <li>• \$ – Zeichenfolgenbefehle</li> <li>• l — List-Befehle</li> <li>• s — Set-Befehle</li> <li>• h — Hash-Befehle</li> <li>• z — Sortierte Set-Befehle</li> <li>• x — Abgelaufene Ereignisse (Ereignisse, die jedes Mal generiert werden, wenn ein Schlüssel abläuft)</li> <li>•</li> </ul>

Name	Details	Beschreibung
		<p>e — Ausgeschiedene Ereigniss e (Ereignisse, die generiert werden, wenn ein Schlüssel für maxmemory entfernt wird)</p> <ul style="list-style-type: none"> <li>• A — Ein Alias für g\$!shzxe</li> </ul> <p>Es kann eine beliebige Kombinati on dieser Ereignistypen vorliegen. AKEBedeutet beispielsweise, dass Redis Benachrichtigungen aller Ereignistypen veröffentlichen OSS kann.</p> <p>Verwenden Sie nur die oben aufgelisteten Zeichen. Die versuchte Eingabe anderer Zeichen führt zu Fehlermeldungen.</p> <p>Für diesen Parameter ist standardmäßig eine leere Zeichenfolge eingestellt. Dies bedeutet, dass die Keyspace- Ereignisbenachrichtigung deaktive rt ist.</p>

Name	Details	Beschreibung
repl-backlog-size	Standard: 1048576 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	<p>Die Größe, in Byte, des Backlogs des primären Knotens. Das Backlog dient zur Aufzeichnung von Aktualisierungen an den Daten auf dem primären Knoten. Wenn sich eine Read Replica mit dem primären Knoten verbindet, versucht sie, eine teilweise Synchronisierung (psync) durchzuführen. Dabei wendet sie die Daten aus dem Backlog an, um den primären Knoten auf den neuesten Stand zu bringen. Wenn psync fehlschlägt, ist eine vollständige Synchronisierung erforderlich.</p> <p>Der Mindestwert für diesen Parameter ist 16384.</p> <div data-bbox="1008 1146 1511 1509"><p> <b>Note</b></p><p>Ab Redis OSS 2.8.22 gilt dieser Parameter sowohl für den primären Cluster als auch für die Read Replicas.</p></div>

Name	Details	Beschreibung
<code>repl-backlog-ttl</code>	<p>Standard: 3600</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Anzahl von Sekunden, für die der primäre Knoten den Backlog-Puffer beibehält. Ab dem Zeitpunkt, an dem die Verbindung des letzte Replikatknotens getrennt wurde, bleiben die Daten im Backlog intakt, bis <code>repl-backlog-ttl</code> abläuft. Wenn das Replikat innerhalb dieses Zeitraums keine Verbindung zum primären Knoten hergestellt hat, dann gibt der primäre Knoten den Backlog-Puffer frei. Wenn das Replikat sich schließlich verbindet, muss es sich einer vollständigen Synchronisierung mit dem primären Knoten unterziehen.</p> <p>Wenn dieser Parameter auf „0“ eingestellt ist, wird der Backlog-Puffer niemals freigegeben.</p>

Name	Details	Beschreibung
<code>repl-timeout</code>	Standard: 60 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Stellt den Zeitraum für die Zeitüberschreitung in Sekunden dar für: <ul style="list-style-type: none"> <li>• Massendatenübertragung während der Synchronisierung, aus der Perspektive des Read Replica</li> <li>• Zeitüberschreitung des primären Knotens aus der Perspektive des Replikats</li> <li>• Zeitüberschreitung des Replikats aus der Perspektive des primären Knotens</li> </ul>

### Redis 2.6.13-Parameter OSS

Familie der Parametergruppe: redis2.6

Redis OSS 2.6.13 war die erste Version von Redis, die von unterstützt wurde. OSS ElastiCache Die folgende Tabelle zeigt die unterstützten Redis OSS 2.6.13-Parameter. ElastiCache

Name	Details	Beschreibung
<code>activeresharding</code>	Standard: yes Typ: Zeichenfolge (yes/no) Veränderbar: Ja Änderungen finden statt: Bei der Erstellung	Bestimmt, ob die aktive Rehashing-Funktion von Redis aktiviert werden soll. Die Haupt-Hash-Tabelle wird zehnmal pro Sekunde erneut gehasht; jeder Rehash-Vorgang benötigt 1 Millisekunde Zeit. CPU  Es empfiehlt sich, diesen Wert gering zu halten. Wenn einem Cluster eine neue Parametergruppe zugewiesen wird, muss dieser Wert in

Name	Details	Beschreibung
appendonly	<p>Standard: no</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>der alten und in der neuen Parametergruppe identisch sein.</p> <p>Aktiviert oder deaktiviert die Redis-Funktion „Nur Dateien anhängen“ (). AOF erfasst alle OSS Redis-Befehle, die Daten im Cache ändern, und wird zur Wiederherstellung nach bestimmten Knotenausfällen verwendet.</p> <p>Der Standardwert ist nein, was bedeutet, dass er ausgeschaltet AOF ist. Setzen Sie diesen Parameter auf Ja, um ihn zu aktivieren AOF.</p> <p>Weitere Informationen finden Sie unter <a href="#">Minimieren von Ausfällen</a>.</p> <div data-bbox="829 909 1507 1272"><p> <b>Note</b></p><p>Append Only Files (AOF) wird für die Knoten cache.t1.micro und cache.t2.* nicht unterstützt. Für Knoten dieser Art wird der Parameterwert appendonly ignoriert.</p></div> <div data-bbox="829 1371 1507 1591"><p> <b>Note</b></p><p>Für Multi-AZ-Replikationsgruppen ist dies nicht zulässig. AOF</p></div>

Name	Details	Beschreibung
<code>appendfsync</code>	<p>Standard: <code>everysec</code></p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Wenn auf <code>yes</code> gesetzt <code>appendonly</code> ist, steuert, wie oft der AOF Ausgabepuffer auf die Festplatte geschrieben wird:</p> <ul style="list-style-type: none"> <li><code>no</code> – der Puffer wird nach Bedarf auf die Festplatte geschrieben.</li> <li><code>everysec</code> – Der Puffer wird einmal pro Sekunde geleert. Dies ist die Standardinstellung.</li> <li><code>always</code> (immer) – der Puffer wird jedes Mal geleert, wenn Daten im Cluster geändert werden.</li> <li><code>Appendfsync</code> wird ab der Version 2.8.22 nicht mehr unterstützt.</li> </ul>
<code>client-output-buffer-limit-normal-hard-limit</code>	<p>Standard: 0</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Bytes erreicht, wird die Verbindung des Clients getrennt. Der Standard ist null (kein festes Limit).</p>
<code>client-output-buffer-limit-normal-soft-limit</code>	<p>Standard: 0</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Bytes erreicht, wird die Verbindung des Clients getrennt, aber nur dann, wenn diese Bedingung <code>client-output-buffer-limit-normal-soft-seconds</code> lang andauert. Der Standard ist null (kein weiches Limit).</p>

Name	Details	Beschreibung
<code>client-output-buffer-limit-normal-soft-seconds</code>	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Wenn der Ausgabepuffer eines Clients länger als die angegebene Anzahl von Sekunden bei <code>client-output-buffer-limit-normal-soft-limit</code> Bytes verbleibt, wird die Verbindung des Clients getrennt. Der Standard ist null (kein Zeitlimit).
<code>client-output-buffer-limit-pubsub-hard-limit</code>	Standard: 33554432 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Für OSS Publish/Subscribe-Clients von Redis: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird die Verbindung zum Client getrennt.
<code>client-output-buffer-limit-pubsub-soft-limit</code>	Standard: 8388608 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Für OSS Publish/Subscribe-Clients von Redis: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird die Verbindung zum Client getrennt, aber nur, wenn dieser Zustand weiterhin besteht. <code>client-output-buffer-limit-pubsub-soft-seconds</code>
<code>client-output-buffer-limit-pubsub-soft-seconds</code>	Standard: 60 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Für OSS Publish/Subscribe-Clients von Redis: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden auf <code>client-output-buffer-limit-pubsub-soft-limit</code> Byte verbleibt, wird die Verbindung zum Client getrennt.

Name	Details	Beschreibung
<code>client-output-buffer-limit-slave-hard-limit</code>	<p>Standard: Für Werte siehe <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a></p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p>	Für OSS Redis-Read-Replicas: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird die Verbindung zum Client getrennt.
<code>client-output-buffer-limit-slave-soft-limit</code>	<p>Standard: Für Werte siehe <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a></p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p>	Für OSS Redis-Read-Replicas: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Byte erreicht, wird die Verbindung zum Client getrennt, aber nur, wenn dieser Zustand weiterhin besteht. <code>client-output-buffer-limit-slave-soft-seconds</code>
<code>client-output-buffer-limit-slave-soft-seconds</code>	<p>Standard: 60</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p>	Für OSS Redis-Read-Replicas: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden auf <code>client-output-buffer-limit-slave-soft-limit</code> Byte verbleibt, wird die Verbindung zum Client getrennt.
<code>databases</code>	<p>Standard: 16</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>Änderungen finden statt: Bei der Erstellung</p>	<p>Die Anzahl der logischen Partitionen, in die die Datenbanken aufgeteilt sind. Es empfiehlt sich, diesen Wert gering zu halten.</p> <p>Es empfiehlt sich, diesen Wert gering zu halten. Wenn einem Cluster eine neue Parametergruppe zugewiesen wird, muss dieser Wert in der alten und in der neuen Parametergruppe identisch sein.</p>

Name	Details	Beschreibung
hash-max-ziplist-entries	Standard: 512 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Hashes verwendete Speichermenge. Hashes mit weniger als der angegebenen Anzahl von Einträgen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
hash-max-ziplist-value	Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Hashes verwendete Speichermenge. Hashes mit kleineren Einträgen als die angegebene Anzahl von Bytes werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
list-max-ziplist-entries	Standard: 512 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Listen verwendete Speichermenge. Listen mit weniger als der angegebenen Anzahl von Einträgen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
list-max-ziplist-value	Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für Listen verwendete Speichermenge. Listen mit kleineren Einträgen als die angegebene Anzahl von Bytes werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.

Name	Details	Beschreibung
<code>lua-time-limit</code>	<p>Standard: 5000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p>	<p>Die maximale Ausführungszeit für ein Lua-Skript in Millisekunden, bevor ElastiCache Maßnahmen ergriffen werden, um das Skript zu beenden.</p> <p>Wenn sie überschritten <i>lua-time-limit</i> wird, geben alle OSS Redis-Befehle einen Fehler in der Form ____- zurück. BUSY Da dieser Status viele wichtige OSS Redis-Operationen stören kann, ElastiCache wird zunächst ein Befehl ausgegeben. SCRIPTKILL Wenn dies nicht erfolgreich ist, ElastiCache wird Redis zwangsweise neu gestartet. OSS</p>
<code>maxclients</code> – Dieser Wert gilt für alle Instance-Typen mit Ausnahme der explizit angegebenen.	<p>Standard: 65000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>t2.medium Standard: 20 000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>t2.small Standard: 20 000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>t2.micro Standard: 20 000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p>	<p>Die maximale Anzahl von Clients, die zu jedem beliebigen Zeitpunkt angeschlossen sein können.</p>

Name	Details	Beschreibung
	<p>t4g.micro Standard: 20 000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>t3.medium Standard: 46 000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>t3.small Standard: 46 000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> <p>t3.micro Standard: 20 000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p>	
maxmemory-policy	<p>Standard: volatile-lru</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Bereinigungsrichtlinie für Schlüssel, wenn die maximale Speichernutzung erreicht ist.</p> <p>Gültige Werte sind: volatile-lru   allkeys-lru   volatile-random   allkeys-random   volatile-ttl   noeviction</p> <p>Weitere Informationen finden Sie unter <a href="#">Verwenden von Valkey oder OSS Redis</a> als Cache. LRU</p>

Name	Details	Beschreibung
maxmemory-samples	Standard: 3 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bei Berechnungen least-recently-used (LRU) und time-to-live (TTL) stellt dieser Parameter die Stichprobengröße der zu überprüfenden Schlüssel dar. Standardmäßig OSS wählt Redis 3 Schlüssel aus und verwendet den Schlüssel, der zuletzt verwendet wurde.
reserved-memory	Standard: 0 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	<p>Der Gesamtspeicher, in Byte, der nicht für Daten verwendet werden soll. Standardmäßig wächst der OSS Redis-Knoten so lange, bis er die des Nodes verbraucht maxmemory (siehe). <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a> In diesem Fall wird die Knotenleistung aufgrund der übermäßigen Speicherauslagerung beeinträchtigt. Indem Sie Speicher reservieren, können Sie einen Teil des verfügbaren Speichers für andere OSS Zwecke als Redis reservieren, um den Umfang des Paging zu reduzieren.</p> <p>Dieser Parameter ist spezifisch für die ElastiCache Redis-Standarddistribution und nicht Teil der Standarddistribution. OSS</p> <p>Weitere Informationen erhalten Sie unter <a href="#">reserved-memory-percent</a> und <a href="#">Verwaltung des reservierten Speichers für Valkey und Redis OSS</a>.</p>

Name	Details	Beschreibung
<code>set-max-intset-entries</code>	<p>Standard: 512</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	Bestimmt die für bestimmte Arten von Sätzen verwendete Speichermenge (Zeichenfolgen, die in Radix 10 Ganzzahlen im Bereich von signierten 64-Bit-Ganzzahlen sind). Solche Sätze mit weniger als der angegebenen Anzahl von Einträgen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
<code>slave-allow-chaining</code>	<p>Standard: no</p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Nein</p>	Bestimmt, ob eine Read Replica in Redis eigene Read Replicas haben OSS kann.
<code>slowlog-log-slower-than</code>	<p>Standard: 10000</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	Die maximale Ausführungszeit in Mikrosekunden für Befehle, die von der OSS Redis Slow Log-Funktion protokolliert werden.
<code>slowlog-max-len</code>	<p>Standard: 128</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	Die maximale Länge des Redis OSS Slow Log.

Name	Details	Beschreibung
tcp-keepalive	<p>Standard: 0</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Wenn dies auf einen Wert ungleich null (N) eingestellt wird, werden Knoten-Clients alle N Sekunden abgefragt, um sicherzustellen, dass sie noch verbunden sind. Bei der Standardinstellung „0“ finden keine solche Abfragen statt.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>⚠ Important</b></p> <p>Einige Aspekte dieses Parameters wurden in Redis OSS Version 3.2.4 geändert. Siehe <a href="#">Die Parameter wurden in Redis OSS 3.2.4 geändert (verbessert)</a>.</p> </div>
timeout	<p>Standard: 0</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p>	<p>Die Anzahl von Sekunden, die ein Knoten wartet, bevor eine Zeitüberschreitung eintritt. Die Werte sind:</p> <ul style="list-style-type: none"> <li>• 0 – Die Verbindung eines ungenutzten Client niemals trennen.</li> <li>• 1-19 – Ungültige Werte.</li> <li>• &gt;=20 – Die Anzahl von Sekunden, die ein Knoten wartet, bevor die Verbindung eines ungenutzten Clients getrennt wird.</li> </ul>

Name	Details	Beschreibung
<code>zset-max-ziplist-entries</code>	Standard: 128 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für sortierte Sätze verwendete Speichermenge. Sortierte Sätze mit weniger als der angegebenen Anzahl von Elementen werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.
<code>zset-max-ziplist-value</code>	Standard: 64 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort	Bestimmt die für sortierte Sätze verwendete Speichermenge. Sortierte Sätze mit kleineren Einträgen als die angegebene Anzahl von Bytes werden mit besonderer Codierung gespeichert, durch die Platz eingespart wird.

#### Note

Wenn Sie keine Parametergruppe für Ihren Redis OSS 2.6.13-Cluster angeben, wird eine Standardparametergruppe (`default.redis2.6`) verwendet. Sie können die Werte der Parameter in der Parametergruppe nicht ändern. Sie können jedoch jederzeit immer eine benutzerdefinierte Parametergruppe erstellen und Sie Ihrem Cluster zuweisen.

## OSSRedis-Knotentyp-spezifische Parameter

Obwohl die meisten Parameter über einen einzelnen Wert verfügen, ist bei einigen Parametern der jeweilige Wert vom verwendeten Knotentyp abhängig. Die folgende Tabelle zeigt die Standardwerte der Parameter `maxmemory`, `client-output-buffer-limit-slave-hard-limit` und `client-output-buffer-limit-slave-soft-limit` für jeden Knotentyp. Der Wert `maxmemory` ist die maximale Anzahl von Bytes, die für Ihre Verwendung, für Daten oder für andere Zwecke auf dem Knoten verfügbar sind. Weitere Informationen finden Sie unter [Verfügbarer Speicher](#).

**Note**

Der Parameter `maxmemory` kann nicht geändert werden.

Knotentyp	Maxmemory	Client-output-buffer-limitC - 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
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

Knotentyp	Maxmemory	lient-output-buffer-limitC - slave-hard-limit	C lient-output-buffe r-limit - slave-soft-limit
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
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

Knotentyp	Maxmemory	Client-output-buffer-limit - slave-hard-limit	Client-output-buffer-limit - slave-soft-limit
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
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

Knotentyp	Maxmemory	Client-output-buffer-limit - slave-hard-limit	Client-output-buffer-limit - slave-soft-limit
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
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

Knotentyp	Maxmemory	lient-output-buffer-limitC - slave-hard-limit	C lient-output-buffe r-limit - slave-soft-limit
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
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**

Alle Instance-Typen der aktuellen Generation werden standardmäßig in einer Amazon Virtual Private Cloud VPC erstellt.

T1-Instances unterstützen Multi-AZ nicht.

T1- und T2-Instances unterstützen Redis OSS AOF nicht.

OSSRedis-Konfigurationsvariablen `appendonly` und `appendfsync` werden in OSS Redis-Version 2.8.22 und höher nicht unterstützt.

## Memcached-spezifische Parameter

### Memcached

Wenn Sie für Ihren Memcached-Cluster keine Parametergruppe angeben, wird eine für Ihre Engine-Version geeignete Standard-Parametergruppe verwendet. Sie können die Werte von Parametern in einer Standard-Parametergruppe nicht ändern. Sie können jedoch jederzeit eine benutzerdefinierte Parametergruppe erstellen und sie Ihrem Cluster zuordnen. Weitere Informationen finden Sie unter [Eine ElastiCache Parametergruppe erstellen](#).

### Themen

- [Änderungen an Memcached 1.6.17](#)
- [In Memcached 1.6.6 hinzugefügte Parameter](#)
- [Parameteränderungen für Memcached 1.5.10](#)
- [In Memcached 1.4.34 hinzugefügte Parameter](#)
- [In Memcached 1.4.33 hinzugefügte Parameter](#)
- [In Memcached 1.4.24 hinzugefügte Parameter](#)
- [In Memcached 1.4.14 hinzugefügte Parameter](#)
- [In Memcached 1.4.5 unterstützte Parameter](#)
- [Overhead von Memcached-Verbindungen](#)
- [Spezifische Parameter des Memcached-Knotentyps](#)

### Änderungen an Memcached 1.6.17

Ab Memcached 1.6.17 unterstützen wir die folgenden administrativen Befehle nicht mehr: `lru_crawler`, `lru` und `slabs`. Mit diesen Änderungen können Sie `lru_crawler` zur

Laufzeit nicht mehr über Befehle aktivieren/deaktivieren. Bitte aktivieren/deaktivieren Sie `lru_crawler`, indem Sie Ihre benutzerdefinierte Parametergruppe ändern.

In Memcached 1.6.6 hinzugefügte Parameter

Für Memcached 1.6.6 werden keine zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.6`

Parameteränderungen für Memcached 1.5.10

Für Memcached 1.5.10 werden die folgenden zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.5`

Name	Details	Beschreibung
<code>no_modern</code>	<p>Standard: 1</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Zulässige Werte: 0, 1</p> <p>Änderungen werden wirksam: Beim Start</p>	<p>Ein Alias für die Deaktivierung von <code>slab_reassign</code> Befehlen, <code>lru_maintainer_thread</code> <code>lru_segmented</code> , und <code>maxconns_fast</code></p> <p>Wenn Sie Memcached 1.5 und höher verwenden, wird der <code>hash_algorithm no_modern</code> auch auf gesetzt. <code>jenkins</code></p> <p>Bei Verwendung von Memcached 1.5.10 wird dies außerdem durch den Parameter gesteuert <code>.inline_ascii_reponse</code> <code>parallelly</code> Das heißt, wenn deaktiviert ist, <code>no_modern</code> ist es auch deaktiviert. <code>inline_ascii_reponse</code> Ab Memcached Engine 1.5.16 gilt der <code>inline_as</code></p>

Name	Details	Beschreibung
		<p><code>no_modern</code> Parameter nicht mehr, sodass die Aktivierung oder Deaktivierung keine Auswirkung auf <code>inline_ascii_response</code> hat.</p> <p>Wenn deaktiviert <code>no_modern</code> ist, dann, <code>slab_reassign_lru_maintainer_thread</code>, <code>lru_segmented</code> und <code>maxconns_fast</code> aktiviert sein. Da es sich bei den <code>hash_algorithm SWITCH</code> Parametern <code>slab_automove</code> und nicht um Parameter handelt, basiert ihre Einstellung auf den Konfigurationen in der Parametergruppe.</p> <p>Wenn Sie den Parameter deaktivieren <code>no_modern</code> und zu ihm zurückkehren möchten, müssen Sie eine benutzerdefinierte Parametergruppe konfigurieren, um diesen Parameter zu deaktivieren, und dann einen Neustart durchführen, damit diese Änderungen wirksam werden.</p> <div data-bbox="1008 1644 1507 1871" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> <b>Note</b></p> <p>Der Standardkonfigurationswert für diesen Parameter wurde am</p> </div>

Name	Details	Beschreibung
		<p>20. August 2021 von 0 in 1 geändert. Der aktualisierte Standardwert wird nach dem 20. August 2021 automatisch von neuen ElastiCache Benutzern für jede Region übernommen. Bestehende ElastiCache Benutzer in den Regionen vor dem 20. August 2021 müssen ihre benutzerdefinierten Parametergruppen manuell ändern, um diese neue Änderung zu übernehmen.</p>
<code>inline_ascii_resp</code>	<p>Standard: 0</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Zulässige Werte: 0, 1</p> <p>Änderungen werden wirksam: Beim Start</p>	<p>Speichert Zahlen der VALUE-Antwort innerhalb eines Elements mit bis zu 24 Bytes. Kleine Verlangsamung für ASCIIget, faster Sätze.</p>

Für Memcached 1.5.10 werden die folgenden Parameter entfernt.

Name	Details	Beschreibung
<code>expirezero_does_no_t_evict</code>	<p>Standard: 0</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p>	<p>Wird in dieser Version nicht länger unterstützt</p>

Name	Details	Beschreibung
	Zulässige Werte: 0, 1  Änderungen werden wirksam: Beim Start	
<code>modern</code>	Standard: 1  Typ: Boolesch  Anpassbar: Ja (erfordert Neustart, wenn auf <code>no_modern</code> gesetzt)  Zulässige Werte: 0, 1  Änderungen werden wirksam: Beim Start	Wird in dieser Version nicht länger unterstützt Ab dieser Version ist <code>no-modern</code> standardmäßig bei jedem Start oder Neustart aktiviert.

In Memcached 1.4.34 hinzugefügte Parameter

Für Memcached 1.4.34 werden keine zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.4`

In Memcached 1.4.33 hinzugefügte Parameter

Für Memcached 1.4.33 werden die folgenden zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.4`

Name	Details	Beschreibung
<code>modern</code>	Standard: aktiviert  Typ: Boolesch  Veränderbar: Ja	Ein Alias für mehrere Funktionen. Das Aktivieren von <code>modern</code> ist gleichbedeutend mit dem Einschalten der folgenden Befehle und der Verwendung eines <code>murmur3-</code>

Name	Details	Beschreibung
	Änderungen werden wirksam: Beim Start	Hash-Algorithmus: <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> und <code>hash_algorithm=murmur3</code> .
watch	<p>Standard: aktiviert</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p> <p>Protokolle werden möglicherweise verworfen, wenn der Benutzer die Grenzwerte für <code>watcher_logbuf_size</code> und <code>worker_logbuf_size</code> erreicht.</p>	Protokollabrufe, Bereinigungen oder Mutationen. Wenn der Benutzer z. B. <code>watch</code> aktiviert, kann er Protokolle sehen, wenn <code>get</code> , <code>set</code> , <code>delete</code> oder <code>update</code> auftreten.
idle_timeout	<p>Standard: 0 (deaktiviert)</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p>	Die minimale Anzahl von Sekunden, die sich ein Client im Leerlauf befinden kann, bevor er zum Schließen aufgefordert wird. Wertebereich: 0 bis 86400.

Name	Details	Beschreibung
<code>track_sizes</code>	<p>Standard: deaktiviert</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p>	<p>Zeigt die Größen an, die jede Slab-Gruppe verbraucht hat.</p> <p>Wenn <code>track_sizes</code> aktiviert ist, können Sie <code>stats sizes</code> ausführen, ohne <code>stats sizes_enable</code> ausführen zu müssen.</p>
<code>watcher_logbuf_size</code>	<p>Standard: 256 (KB)</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p>	<p>Der Befehl <code>watch</code> schaltet die Stream-Protokollierung für Memcached ein. <code>watch</code> kann jedoch Protokolle verwerfen, wenn die Rate der Bereinigungen, Mutationen oder Abrufe so hoch ist, dass der Protokollierungspuffer voll wird. In solchen Fällen kann der Benutzer die Puffergröße erhöhen, um die Gefahr von Protokollverlusten zu verringern.</p>
<code>worker_logbuf_size</code>	<p>Standard: 64 (KB)</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p>	<p>Der Befehl <code>watch</code> schaltet die Stream-Protokollierung für Memcached ein. <code>watch</code> kann jedoch Protokolle verwerfen, wenn die Rate der Bereinigungen, Mutationen oder Abrufe so hoch ist, dass der Protokollierungspuffer voll wird. In solchen Fällen kann der Benutzer die Puffergröße erhöhen, um die Gefahr von Protokollverlusten zu verringern.</p>

Name	Details	Beschreibung
<code>slab_chunk_max</code>	Standard: 524288 (Byte)  Typ: Ganzzahl  Veränderbar: Ja  Änderungen werden wirksam: Beim Start	Gibt die maximale Größe eines Slabs an. Bei Festlegen einer kleineren Slab-Größe wird der Speicher effizienter genutzt. Elemente, die größer als <code>slab_chunk_max</code> sind, werden über mehrere Slabs hinweg aufgeteilt.
<code>lru_crawler metadump [all 1 2 3]</code>	Standard: deaktiviert  Typ: Boolesch  Veränderbar: Ja  Änderungen werden wirksam: Sofort	Wenn <code>lru_crawler</code> aktiviert ist, sichert dieser Befehl ein Abbild aller Schlüssel.  <code>all 1 2 3</code> – alle Slabs oder geben Sie eine bestimmte Slab-Nummer an

### In Memcached 1.4.24 hinzugefügte Parameter

Für Memcached 1.4.24 werden die folgenden zusätzlichen Parameter unterstützt.

Familie der Parametergruppe: `memcached1.4`

Name	Details	Beschreibung
<code>disable_flush_all</code>	Standard: 0 (deaktiviert)  Typ: Boolesch  Veränderbar: Ja  Änderungen werden wirksam: Beim Start	Hinzufügen von Parameter (-F), um <code>flush_all</code> zu deaktivieren. Nützlich, wenn Sie niemals eine volle Leerung für Produktions-Instances ausführen möchten.

Name	Details	Beschreibung
		Werte: 0, 1 (Benutzer kann einen <code>flush_all</code> ausführen, wenn der Wert 0 ist).
hash_algorithm	Standard: jenkins Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Beim Start	Der zu verwendende Hash-Algorithmus. Zulässige Werte: murmur3 und jenkins.

Name	Details	Beschreibung
<p><code>lru_crawler</code></p>	<p>Standard: 0 (deaktiviert)</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Nach dem Neustart</p> <div data-bbox="651 684 971 1430" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Sie können <code>lru_crawler</code> vorübergehend zur Laufzeit über die Befehlszeile aktivieren. Weitere Informationen finden Sie in der Spalte „Description“.</p> </div>	<p>Bereinigt Slab-Klassen abgelaufener Elemente. Dies ist ein Vorgang mit geringen Auswirkungen, der im Hintergrund ausgeführt wird. Er erfordert zurzeit, dass das Durchsuchen mit einem manuellen Befehl initiiert wird.</p> <p>Um ihn auf bestimmte Zeit zu aktivieren, führen Sie <code>lru_crawler enable</code> über die Befehlszeile aus.</p> <p><code>lru_crawler 1,3,5</code> durchläuft Slab-Klassen 1, 3 und 5 auf der Suche nach abgelaufenen Elementen, die zur Freelist hinzugefügt werden sollen.</p> <p>Werte: 0,1</p> <div data-bbox="1011 1194 1508 1852" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Wenn <code>lru_crawler</code> über die Befehlszeile aktiviert wird, wird der Crawler so lange aktiviert, bis er über die Befehlszeile oder beim nächsten Neustart deaktiviert wird. Um ihn permanent zu aktivieren, müssen Sie den Parameterwert ändern. Weitere Informationen finden Sie unter</p> </div>

Name	Details	Beschreibung
		<p><a href="#">Änderung einer ElastiCache Parametergruppe.</a></p>
<code>lru_maintainer</code>	<p>Standard: 0 (deaktiviert)</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p>	<p>Ein Hintergrund-Thread, der Elemente zwischen den Objekten hin und her schiebt, LRUs sobald die Kapazitäten erreicht sind.</p> <p>Werte: 0, 1.</p>
<code>expirezero_does_not_evict</code>	<p>Standard: 0 (deaktiviert)</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p>	<p>Bewirkt bei Verwendung zusammen mit <code>lru_maintainer</code>, dass Elemente mit der Ablaufzeit 0 nicht bereinigt werden können.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Warning</b></p> <p>Dadurch kann der Speicher für andere Elemente, die bereinigt werden können, knapp werden.</p> </div> <p>Kann so eingestellt werden, dass <code>lru_maintainer</code> ignoriert wird.</p>

In Memcached 1.4.14 hinzugefügte Parameter

Für Memcached 1.4.14 werden die folgenden zusätzlichen Parameter unterstützt.

## Familie der Parametergruppe: memcached1.4

## In Memcached 1.4.14 hinzugefügte Parameter

Name	Beschreibung
<code>config_max</code>	Die maximale Anzahl von ElastiCache Konfigurationseinträgen.
<code>config_size_max</code>	Die maximale Größe der Konfigurationseinträge in Byte.
<code>hashpower_init</code>	Die Anfangsgröße der ElastiCache Hashtabelle, ausgedrückt als Zweierpotenz. Der Standard ist 16 ( $2^{16}$ ), oder 65536 Schlüssel.
<code>maxconns_fast</code>	Ändert, auf welche Art neue Verbindungsanforderungen gehandhabt werden, wenn das maximale Verbindungslimit erreicht ist. Wenn dieser Parameter auf 0 (Null) eingestellt ist, werden neue Verbindungen zur Rückstandwarteschlange hinzugefügt und gewartet, bis andere Verbindungen geschlossen werden. Wenn der Parameter auf 1

Name	Beschreibung
	gesetzt ist, wird ein Fehler an den Client ElastiCache gesendet und die Verbindung sofort geschlossen.
slab_automove	Korrigiert den Slab-AutoMove-Algorithmus: Wenn dieser Parameter auf 0 (Null) eingestellt ist, ist der AutoMove-Algorithmus deaktiviert. Wenn er auf 1 gesetzt ist, ElastiCache wird beim automatischen Verschieben von Platten ein langsamer, konservativer Ansatz gewählt. Wenn der Wert auf 2 gesetzt ist, werden Platten bei jeder Räumung ElastiCache aggressiv bewegt. (Dieser Modus wird ausschließlich zu Testzwecken empfohlen.)

Name	Beschreibung
slab_reassign	Aktiviert oder deaktiviert die Slab-Neuzuweisung. Wenn dieser Parameter auf 1 eingestellt ist, können Sie mit dem Befehl „slabs reassign“ Speicher manuell neu zuweisen.

### In Memcached 1.4.5 unterstützte Parameter

Familie der Parametergruppe: memcached1.4

Für Memcached 1.4.5 werden die folgenden Parameter unterstützt.

### In Memcached 1.4.5 hinzugefügte Parameter

Name	Details	Beschreibung
backlog_queue_limit	Standard: 1024 Typ: Ganzzahl Veränderbar: Nein	Das Limit der Rückstandwarteschlange (Backlog Queue).
binding_protocol	Standard: auto Typ: Zeichenfolge	Das Bindungsprotokoll. Zulässige Werte sind: <code>ascii</code> und <code>auto</code> .

Name	Details	Beschreibung
	Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Eine Anleitung zum Ändern des Wertes von <code>binding_protocol</code> finden Sie unter <a href="#">Änderung einer ElastiCache Parametergruppe</a> .
<code>cas_disabled</code>	Standard: 0(false) Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Bei einem Wert von 1 (true) werden die Operationen check und set (CAS) deaktiviert und gespeicherte Elemente verbrauchen 8 Byte weniger als bei aktivierter Option. CAS
<code>chunk_size</code>	Standard: 48 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Die Mindestmenge, in Byte, von Speicher, die für den Schlüssel, Wert und die Flags des kleinsten Elements zuzuordnen sind.
<code>chunk_size_growth_factor</code>	Standard: 1.25 Typ: Gleitkommazahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Der Wachstumsfaktor, der die Größe aller nachfolgenden Memcached-Datenblöcke steuert. Jeder Datenblock wird <code>chunk_size_growth_factor</code> Mal größer als der vorherige Datenblock sein.
<code>error_on_memory_exhausted</code>	Standard: 0(false) Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Bei dem Wert 1 (true) gibt Memcached einen Fehler zurück, wenn kein Speicher zum Speichern der Elemente mehr verfügbar ist, anstatt Elemente zu bereinigen.

Name	Details	Beschreibung
large_memory_pages	Standard: 0(false) Typ: Boolesch Veränderbar: Nein	Falls 1 (true), ElastiCache wird versucht, große Speicherseiten zu verwenden.
lock_down_paged_memory	Standard: 0(false) Typ: Boolesch Veränderbar: Nein	Falls 1 (wahr), ElastiCache wird der gesamte ausgelagerte Speicher gesperrt.
max_item_size	Standard: 1048576 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Die Größe, in Byte, des größten Elements, das im Cluster gespeichert werden kann.
max_simultaneous_connections	Standard: 65000 Typ: Ganzzahl Veränderbar: Nein	Die maximale Anzahl gleichzeitiger Verbindungen.
maximize_core_file_limit	Standard: 0(false) Typ: Boolesch Anpassbar: Änderungen werden wirksam: Nach dem Neustart	Falls 1 (wahr), ElastiCache wird das Limit für die Kerndateien maximiert.

Name	Details	Beschreibung
<code>memcached_connections_overhead</code>	Standard: 100 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Nach dem Neustart	Die Menge an Speicher, die für Memcached-Verbindungen und verschiedene Overhead-Anforderungen zu reservieren ist. Informationen zu diesem Parameter finden Sie unter <a href="#">Overhead von Memcached-Verbindungen</a> .
<code>requests_per_event</code>	Standard: 20 Typ: Ganzzahl Veränderbar: Nein	Die maximale Anzahl von Anforderungen pro Ereignis für eine bestimmte Verbindung. Dieses Limit wird benötigt, um zu verhindern, dass die Ressourcen aufgebraucht werden.

## Overhead von Memcached-Verbindungen

Der auf jedem Knoten zum Speichern von Elementen verfügbar gemachte Speicher ist der verfügbare Gesamtspeicher auf diesem Knoten (der im Parameter `max_cache_memory` gespeichert wird) minus des für Verbindungen und anderes Overhead verwendeten Speichers (der im Parameter `memcached_connections_overhead` gespeichert wird). Ein Knoten vom Typ `cache.m1.small` verfügt z. B. über `max_cache_memory` von 1300 MB. Bei einem `memcached_connections_overhead`-Standardwert von 100 MB hat der Memcached-Prozess 1200 MB zum Speichern von Elementen verfügbar.

Die Standardwerte für den Parameter `memcached_connections_overhead` genügen für die meisten Anwendungsfälle. Die erforderliche Menge, die für das Overhead der Verbindungen zugewiesen werden muss, kann abhängig von mehreren Faktoren, darunter Anforderungsrate, Nutzlastgröße und die Anzahl von Verbindungen, variieren.

Sie können den Wert von `memcached_connections_overhead` ändern, um den Anforderungen Ihrer Anwendung besser zu genügen. Wenn z. B. der Wert des Parameters `memcached_connections_overhead` erhöht wird, verringert sich die Menge des Speichers, der zum Speichern von Elementen verfügbar ist. Auch wird ein größerer Puffer für das Overhead von Verbindungen geschaffen. Wenn der Wert des Parameters `memcached_connections_overhead` verringert wird, erhalten Sie mehr Speicher zum Speichern von Elementen. Dadurch kann sich aber das Risiko bezüglich Swap-Nutzung und Leistungseinbußen erhöhen. Wenn Swap-

Nutzung und Leistungseinbußen zu beobachten sind, versuchen Sie, den Wert des Parameters `memcached_connections_overhead` zu verringern.

### Wichtig

Für den Knotentyp `cache.t1.micro` wird der Wert für `memcached_connections_overhead` folgendermaßen bestimmt:

- Wenn Ihr Cluster die Standardparametergruppe verwendet, ElastiCache wird der Wert für `memcached_connections_overhead` auf 13 MB gesetzt.
- Wenn Ihr Cluster eine Parametergruppe nutzt, die Sie selbst erstellt haben, können Sie als Wert für `memcached_connections_overhead` einen Wert Ihrer Wahl festlegen.

### Spezifische Parameter des Memcached-Knotentyps

Obwohl die meisten Parameter über einen einzelnen Wert verfügen, ist bei einigen Parametern der jeweilige Wert vom verwendeten Knotentyp abhängig. Die folgende Tabelle zeigt die Standardwerte der Parameter `max_cache_memory` und `num_threads` für jeden Knotentyp. Die Werte für diese Parameter können nicht geändert werden.

Knotentyp	<code>max_cache_memory</code> (in Megabyte)	<code>num_threads</code>
<code>cache.t1.micro</code>	213	1
<code>cache.t2.micro</code>	555	1
<code>cache.t2.small</code>	1588	1
<code>cache.t2.medium</code>	3301	2
<code>cache.t3.micro</code>	512	2
<code>cache.t3.small</code>	1402	2
<code>cache.t3.medium</code>	3364	2
<code>cache.t4g.micro</code>	512	2

Knotentyp	max_cache_memory (in Megabyte)	num_threads
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
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

Knotentyp	max_cache_memory (in Megabyte)	num_threads
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
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

Knotentyp	max_cache_memory (in Megabyte)	num_threads
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
dache.r6g.2xlarge	54084	8
dache.r6g.4xlarge	108347	16
dache.r6g.8xlarge	214577	32
dache.r6g.12xlarge	325400	48
dache.r6g.16xlarge	429154	64
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16

Knotentyp	max_cache_memory (in Megabyte)	num_threads
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

### Note

Alle T2-Instances werden in einer Amazon Virtual Private Cloud (AmazonVPC) erstellt.

## Skalierung ElastiCache

Sie können Ihren ElastiCache Cache an Ihre Bedürfnisse anpassen. Serverlose Caches und selbst entworfene Cluster bieten verschiedene Skalierungsoptionen.

### ElastiCache Serverlose Skalierung

ElastiCache Serverless passt sich automatisch Ihrem Workload-Verkehr an, wenn er steigt oder fällt. Verfolgt für jeden ElastiCache serverlosen Cache ElastiCache kontinuierlich die Auslastung von Ressourcen wie Arbeitsspeicher CPU und Netzwerk. Wenn eine dieser Ressourcen eingeschränkt ist, skaliert ElastiCache Serverless, indem es einen neuen Shard hinzufügt und die Daten auf den neuen Shard umverteilt, ohne dass es zu Ausfallzeiten Ihrer Anwendung kommt. Sie können die von Ihrem Cache-In CloudWatch verbrauchten Ressourcen überwachen, indem Sie die BytesUsedForCache Metrik für den Cache-Datenspeicher und ElastiCacheProcessingUnits () ECPU für die Computennutzung überwachen.

### Festlegen von Skalierungslimits zur Kostenkontrolle

Sie können wählen, ob Sie eine maximale Nutzung sowohl für den Cache-Datenspeicher als auch für ECPU /Sekunde für Ihren Cache konfigurieren möchten, um die Cache-Kosten zu kontrollieren. Dadurch wird sichergestellt, dass Ihre Cache-Nutzung das konfigurierte Maximum nicht überschreitet.

Wenn Sie ein Skalierungsmaximum festlegen, kann es bei Ihrer Anwendung zu einer verminderten Cacheleistung kommen, wenn der Cache das Maximum erreicht. Wenn Sie ein Maximum für den

Cache-Datenspeicher festlegen und Ihr Cache-Datenspeicher das Maximum erreicht, ElastiCache beginnt das Löschen von Daten in Ihrem Cache, für die ein Time-To-Live (TTL) gesetzt ist. Dabei wird die LRU Logik verwendet. Wenn keine Daten vorhanden sind, die gelöscht werden können, wird bei Anfragen zum Schreiben zusätzlicher Daten die Fehlermeldung Out Of Memory () OOM angezeigt. Wenn Sie ein Maximum von ECPU /second festlegen und die Rechenauslastung Ihres Workloads diesen Wert überschreitet, ElastiCache beginnt die Drosselung von Anfragen.

Wenn Sie ein Höchstlimit für BytesUsedForCache oder einrichtenElastiCacheProcessingUnits, empfehlen wir dringend, einen CloudWatch Alarm bei einem niedrigeren Wert als dem Höchstwert einzurichten, damit Sie benachrichtigt werden, wenn Ihr Cache in der Nähe dieser Grenzwerte arbeitet. Wir empfehlen, einen Alarm bei 75 % des von Ihnen festgelegten maximalen Limits einzustellen. Informationen zum Einrichten von CloudWatch Alarmen finden Sie in der Dokumentation.

## Vorskalierung mit Serverless ElastiCache

### ElastiCache Serverlose Vorskalierung

Mit der Vorskalierung, auch Pre-Warming genannt, können Sie die unterstützten Mindestgrenzen für Ihren Cache festlegen. ElastiCache Sie können diese Mindestwerte für ElastiCache Verarbeitungseinheiten (ECPUs) pro Sekunde oder Datenspeicher festlegen. Dies kann bei der Vorbereitung auf erwartete Skalierungsereignisse nützlich sein. Wenn ein Spieleunternehmen beispielsweise innerhalb der ersten Minute, in der sein neues Spiel veröffentlicht wird, eine 5-fache Zunahme der Logins erwartet, kann es seinen Cache für diesen deutlichen Anstieg der Nutzung vorbereiten.

Sie können die Vorskalierung mit der ElastiCache Konsole, oder durchführen. CLI API ElastiCache Serverless aktualisiert den verfügbaren Wert ECPUs pro Sekunde im Cache innerhalb von 60 Minuten und sendet eine Ereignisbenachrichtigung, wenn die Aktualisierung des Mindestlimits abgeschlossen ist.

### So funktioniert die Vorskalierung

Wenn das Mindestlimit für ECPUs /second oder Datenspeicher über die Konsole,, oder API aktualisiert wirdCLI, ist dieses neue Limit innerhalb von 1 Stunde verfügbar. ElastiCache Serverless unterstützt 30.000 ECPUs pro Sekunde bei leerem Cache und bis zu 90.000 ECPUs pro Sekunde bei Verwendung der Funktion „Aus Replikat lesen“. ElastiCache kann den Wert pro Sekunde alle 10 —12 Minuten verdoppeln. ECPUs Diese Skalierungsgeschwindigkeit ist für die meisten Workloads ausreichend. Wenn Sie davon ausgehen, dass ein bevorstehendes Skalierungsereignis diese Rate

überschreiten könnte, empfehlen wir, den Mindestwert pro Sekunde auf den Höchstwert ECPUs pro Sekunde festzulegen, den Sie mindestens 60 Minuten vor dem ECPUs Spitzenereignis erwarten. Andernfalls kann es bei der Anwendung zu erhöhter Latenz und Drosselung von Anfragen kommen.

Sobald die Aktualisierung des Mindestlimits abgeschlossen ist, beginnt ElastiCache Serverless mit der Berechnung des neuen Mindestwerts ECPUs pro Sekunde oder des neuen Mindestspeichers. Dies ist auch dann der Fall, wenn Ihre Anwendung keine Anforderungen im Cache ausführt oder wenn Ihre Datenspeichernutzung unter dem Mindestwert liegt. Wenn Sie den Mindestgrenzwert gegenüber der aktuellen Einstellung herabsetzen, erfolgt die Aktualisierung sofort, sodass ElastiCache Serverless sofort mit der Messung des neuen Mindestlimits beginnt.

### Note

- Wenn Sie ein Mindestnutzungslimit festlegen, wird Ihnen dieses Limit in Rechnung gestellt, auch wenn Ihre tatsächliche Nutzung unter dem Mindestnutzungslimit liegt. ECPUs oder Datenspeichernutzung, die das Mindestnutzungslimit überschreitet, wird der reguläre Tarif berechnet. Wenn Sie beispielsweise ein Mindestnutzungslimit von 100.000 ECPUs €/ Sekunde festlegen, werden Ihnen mindestens 1,224 USD pro Stunde berechnet (unter Verwendung von ECPU Preisen in US-East-1), auch wenn Ihre Nutzung unter dem festgelegten Mindestwert liegt.
- ElastiCache Serverless unterstützt den angeforderten Mindestmaßstab auf aggregierter Ebene im Cache. ElastiCache Serverless unterstützt außerdem maximal 30.000 ECPUs €/ Sekunde pro Steckplatz (90.000 ECPUs €/Sekunde, wenn Read from Replica mithilfe von Verbindungen verwendet wird). READONLY Als bewährte Methode sollte Ihre Anwendung sicherstellen, dass die Schlüsselverteilung zwischen den Valkey- oder OSS Redis-Steckplätzen und der Datenverkehr zwischen den Schlüsseln so einheitlich wie möglich sind.

## Skalierungsgrenzen mithilfe der Konsole festlegen und AWS CLI

### Skalierungsgrenzen mithilfe der AWS Konsole festlegen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich die Engine, die in dem Cache ausgeführt wird, den Sie ändern möchten.

3. Es wird eine Liste der Caches der ausgewählten Engine angezeigt.
4. Wählen Sie den zu ändernden Cache aus, indem Sie das Optionsfeld links neben dem Cache-Namen anklicken.
5. Wählen Sie Aktionen und dann Ändern.
6. Legen Sie unter Nutzungslimits die entsprechenden Speicher - oder Rechenlimits fest.
7. Klicken Sie auf Änderungen in der Vorschau anzeigen und dann auf Änderungen speichern.

## Einstellen von Skalierungsgrenzen mithilfe der AWS CLI

Um Skalierungsgrenzen mit dem zu ändernCLI, verwenden Sie den `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}'
```

## Entfernen von Skalierungsbeschränkungen mit dem CLI

Um Skalierungsgrenzen mithilfe von zu entfernenCLI, setzen Sie die Parameter `Minimum` und `Maximum` auf 0.

Linux:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
ECPUPerSecond={Minimum=0,Maximum=0}'
```

Windows:

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},
ECPUPerSecond={Minimum=0,Maximum=0}'
```

## Skalierung selbst entworfener Cluster

Die Datenmenge, die von einer Anwendung verarbeitet wird, ist selten statisch. Sie steigt und sinkt mit dem Unternehmenswachstum und unterliegt normalen Schwankungen im Bedarf. Wenn Sie Ihren Cache selbst verwalten, müssen Sie für Bedarfsspitzen ausreichend Hardware bereitstellen – dies kann kostspielig sein. Mit Amazon können ElastiCache Sie skalieren, um der aktuellen Nachfrage gerecht zu werden, und zahlen nur für das, was Sie tatsächlich nutzen. ElastiCache ermöglicht es Ihnen, Ihren Cache entsprechend der Nachfrage zu skalieren.

### Note

Wenn ein Valkey- oder OSS Redis-Cluster in einer oder mehreren Regionen repliziert wird, werden diese Regionen der Reihe nach skaliert. Bei der Skalierung werden zuerst die sekundären Regionen und dann die primäre Region skaliert. Beim Herunterskalieren steht die primäre Region an erster Stelle und dann folgen alle sekundären Regionen. Bei der Aktualisierung der Engine-Version lautet die Reihenfolge sekundäre Region und dann primäre Region.

### Themen

- [Skalierung von Clustern für Memcached](#)
- [Skalierung von Clustern für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#)
- [Skalierung von Replikatknoten für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#)
- [Skalierung von Clustern in Valkey oder Redis OSS \(Clustermodus aktiviert\)](#)

## Skalierung von Clustern für Memcached

Die folgenden Tabellen können Ihnen helfen, das richtige Thema für die Skalierungsaktionen zu finden, die Sie durchführen möchten.

### Skalierung von Memcached-Clustern

Memcached-Cluster bestehen aus 1 bis 60 Knoten. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten zu bzw. aus dem Cluster nach oben bzw. nach unten skaliert.

Da Sie in einem Memcached-Cluster Ihre Daten auf alle vorhandenen Knoten verteilen können, müssen Sie selten auf einen Knotentyp mit mehr Speicher hochskalieren. Da Daten in der

Memcached-Engine jedoch nicht beibehalten werden, müssen Sie, wenn Sie auf einen anderen Knotentyp skalieren, Ihren Cluster erstellen, der zunächst leer ist, bis er durch Ihre Anwendung befüllt wird.

### Skalierung von Memcached-Clustern

Aktion	Thema
Ausskalieren	<a href="#">Hinzufügen von Knoten zu einem Cluster</a>
Einskalieren	<a href="#">Löschen von Knoten aus einem Cluster</a>
Ändern von Knotentypen	<a href="#">Vertikale Skalierung von Memcached</a>

Memcached-Cluster bestehen aus 1 bis 60 Knoten. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten zu bzw. aus dem Cluster nach oben bzw. nach unten skaliert.

Da Sie in einem Memcached-Cluster Ihre Daten auf alle vorhandenen Knoten verteilen können, müssen Sie selten auf einen Knotentyp mit mehr Speicher hochskalieren. Da Daten in der Memcached-Engine jedoch nicht beibehalten werden, müssen Sie, wenn Sie auf einen anderen Knotentyp skalieren, Ihren Cluster erstellen, der zunächst leer ist, bis er durch Ihre Anwendung befüllt wird.

### Themen

- [Horizontale Skalierung von Memcached](#)
- [Vertikale Skalierung von Memcached](#)

### Horizontale Skalierung von Memcached

Die Memcached-Engine unterstützt die Partitionierung Ihrer Daten über mehrere Knoten hinweg. Dadurch lassen sich Memcached-Cluster einfach horizontal skalieren. Ein Memcached-Cluster kann aus 1 bis 60 Knoten bestehen. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten horizontal skaliert.

In den folgenden Themen wird erläutert, wie Sie Memcached-Cluster durch Hinzufügen oder Entfernen von Knoten ein- oder ausskalieren.

- [Hinzufügen von Knoten zu einem Cluster](#)
- [Löschen von Knoten aus Ihrem Cluster](#)

Jedes Mal, wenn Sie die Anzahl der Knoten in Ihrem Memcached-Cluster ändern, müssen Sie zumindest einen Teil Ihres Keyspace neu zuordnen, sodass er dem richtigen Knoten zugeordnet ist. Detaillierte Informationen zum Lastausgleich Ihres Memcached-Clusters finden Sie unter [Konfiguration Ihres ElastiCache Clients für einen effizienten Lastenausgleich \(Memcached\)](#).

Wenn Sie für Ihr Memcached-Cluster Auto Discovery verwenden, müssen Sie die Endpunkte in Ihrer Anwendung nicht ändern, wenn Sie Knoten hinzufügen oder entfernen. Weitere Informationen zu Auto Discovery finden Sie unter [Identifizieren Sie automatisch Knoten in Ihrem Cluster \(Memcached\)](#). Wenn Sie Auto Discovery nicht verwenden, müssen Sie bei jeder Änderung der Anzahl der Knoten in Ihrem Memcached-Cluster die Endpunkte in Ihrer Anwendung aktualisieren.

## Vertikale Skalierung von Memcached

Wenn Sie einen Memcached-Cluster aufwärts oder abwärts skalieren, müssen Sie einen neuen Cluster erstellen. Memcached-Cluster sind zunächst leer, bis sie von der Anwendung befüllt werden.

### Important

Vergewissern Sie sich beim Skalieren zu einem kleineren Knotentyp davon, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht. Weitere Informationen finden Sie unter [Auswahl der Cache-Knotengröße](#).

## Themen

- [Vertikale Skalierung von Memcached \(Konsole\)](#)
- [Vertikale Skalierung von Memcached \(AWS CLI\)](#)
- [Vertikale Skalierung von Memcached \(ElastiCache API\)](#)

## Vertikale Skalierung von Memcached (Konsole)

Das folgende Verfahren führt Sie durch die vertikale Skalierung Ihres Clusters mithilfe der ElastiCache Konsole.

### So skalieren Sie einen Memcached-Cluster vertikal (Konsole)

1. Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Die Endpunkte eines Clusters finden \(Konsole\) \(Memcached\)](#).
3. Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Löschen eines neuen Knotens in Memcached](#).

### Vertikale Skalierung von Memcached (AWS CLI)

Nachfolgend wird beschrieben, wie Sie Ihren Memcached-Cache-Cluster mit der AWS CLI vertikal skalieren.

So skalieren Sie einen Memcached-Cache-Cluster vertikal (AWS CLI)

1. Erstellen Sie einen neuen Cache-Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Cluster erstellen mit dem CLI](#).
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Suchen von Endpunkten \(AWS CLI\)](#).
3. Löschen Sie den alten Cache-Cluster. Weitere Informationen finden Sie unter [Verwenden Sie den AWS CLI , um einen ElastiCache Cluster zu löschen](#).

### Vertikale Skalierung von Memcached (ElastiCache API)

Das folgende Verfahren führt Sie durch die vertikale Skalierung Ihres Memcached-Cache-Clusters mithilfe von ElastiCache API

So skalieren Sie einen Memcache-Cache-Cluster vertikal () ElastiCache API

1. Erstellen Sie einen neuen Cache-Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Einen Cluster für Memcached \(\) erstellen ElastiCache API](#)
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cache-Cluster-Endpunkte. Weitere Informationen finden Sie unter [Endpunkte finden \(\) ElastiCache API](#).
3. Löschen Sie den alten Cache-Cluster. Weitere Informationen finden Sie unter [Mit dem ElastiCache API](#).

## Skalierung von Clustern für Valkey oder Redis OSS (Clustermodus deaktiviert)

Bei Valkey- oder Redis-Clustern OSS (Clustermodus deaktiviert) kann es sich um einen Cluster mit einem Knoten mit 0 Shards oder um Cluster mit mehreren Knoten und einem Shard handeln. Cluster mit nur einem Knoten verwenden einen Knoten sowohl für Lese- als auch für Schreibvorgänge. Cluster mit mehreren Knoten haben immer 1 Knoten als Primärknoten mit Lese-/Schreibzugriff mit 0 bis 5 schreibgeschützte Replikationsknoten.

### Themen

- [Skalierung von Einzelknotenclustern für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#)

### Skalierung OSS von Valkey- oder Redis-Clustern

Aktion	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
Einskalieren	<a href="#">Knoten aus einem ElastiCache Cluster entfernen</a>	<a href="#">Skalierung von Clustern in Valkey oder Redis OSS (Clustermodus aktiviert)</a>
Ausskalieren	<a href="#">Hinzufügen von Knoten zu einem Cluster</a>	<a href="#">Online-Resharding für Valkey oder OSS Redis (Clustermodus aktiviert)</a>
Ändern von Knotentypen	<p>Zu einem größeren Knotentyp:</p> <ul style="list-style-type: none"> <li>• <a href="#">Skalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS</a></li> <li>• <a href="#">Skalierung von Valkey- oder Redis-Clustern mit Replikaten OSS</a></li> </ul> <p>Zu einem kleineren Knotentyp:</p> <ul style="list-style-type: none"> <li>• </li> </ul>	<a href="#">Vertikales Online-Skalieren durch Ändern des Knotentyps</a>

Aktion	Valkey oder Redis OSS (Clustermodus deaktiviert)	Valkey oder Redis OSS (Clustermodus aktiviert)
	<a href="#">Herunterskalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS</a> <ul style="list-style-type: none"> <li>• <a href="#">Herunterskalierung von Valkey- oder OSS Redis-Clustern mit Replikaten</a></li> </ul>	
Ändern der Anzahl der Knotengruppen	Wird für Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) nicht unterstützt	<a href="#">Skalierung von Clustern in Valkey oder Redis OSS (Clustermodus aktiviert)</a>

## Inhalt

- [Skalierung von Einzelknotenclustern für Valkey oder Redis OSS \(Clustermodus deaktiviert\)](#)
  - [Skalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS](#)
    - [Skalierung von Clustern mit einem Knoten für Valkey oder Redis OSS \(Clustermodus deaktiviert\) \(Konsole\)](#)
    - [Skalierung von Valkey- oder Redis-Cache-Clustern mit einem Knoten \(\) OSS AWS CLI](#)
    - [Skalierung von Valkey- oder OSS Redis-Cache-Clustern mit einem Knoten \(\) ElastiCache API](#)
  - [Herunterskalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS](#)
    - [Herunterskalierung eines Valkey- oder Redis-Clusters mit einem Knoten \(Konsole\) OSS](#)
    - [Herunterskalierung der Valkey- oder Redis-Cache-Cluster mit einem Knoten \(\) OSS AWS CLI](#)
    - [Herunterskalierung der Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten \(\) ElastiCache API](#)

## Skalierung von Einzelknotenclustern für Valkey oder Redis OSS (Clustermodus deaktiviert)

Valkey- oder Redis-Knoten OSS (Clustermodus deaktiviert) müssen groß genug sein, um alle Cache-Daten plus den Valkey- oder Redis-Overhead aufzunehmen. OSS Um die Datenkapazität

Ihres Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) zu ändern, müssen Sie vertikal skalieren. Sie müssen auf einen größeren Knotentyp skalieren, um die Datenkapazität zu erhöhen, oder auf einen kleineren Knotentyp herunterskalieren, um die Datenkapazität zu reduzieren.

Der ElastiCache Skalierungsprozess ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich beizubehalten, und erfordert eine erfolgreiche Valkey- oder Redis-Replikation. OSS Für Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) empfehlen wir, Valkey oder Redis ausreichend Arbeitsspeicher zur Verfügung zu stellen. OSS

Sie können Ihre Daten nicht auf mehrere Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) partitionieren. Wenn Sie jedoch nur die Lesekapazität Ihres Clusters erhöhen oder verringern müssen, können Sie einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit Replikatknoten erstellen und Read Replicas hinzufügen oder entfernen. Informationen zum Erstellen eines Valkey- oder Redis-Clusters OSS (Clustermodus deaktiviert) mit Replikatknoten unter Verwendung Ihres Valkey- oder OSS Redis-Cache-Clusters mit einem Knoten als primärem Cluster finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

Nachdem Sie den Cluster mit Replikaten erstellt haben, können Sie die Lesekapazität durch Hinzufügen von Read Replicas erweitern. Später kann die Lesekapazität durch Entfernen von Read Replicas wieder reduziert werden. Weitere Informationen finden Sie unter [Erhöhen der Lesekapazität](#) oder [Verringern der Lesekapazität](#).

Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) mit Replikaten bieten nicht nur die Möglichkeit, die Lesekapazität zu skalieren, sondern bieten auch weitere Geschäftsvorteile. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

#### Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder OSS Redis-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

## Themen

- [Skalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS](#)
- [Herunterskalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS](#)

## Skalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS

Wenn Sie einen Valkey- oder OSS Redis-Cluster mit einem Knoten skalieren, führt ElastiCache den folgenden Vorgang durch, unabhängig davon, ob Sie die ElastiCache Konsole, den oder den verwenden. AWS CLI ElastiCache API

1. Es wird ein neuer Cache-Cluster mit demselben Knotentyp in derselben Availability Zone wie der bestehende Cache-Cluster erstellt.
2. Die Cache-Daten im vorhandenen Cache-Cluster werden in den neuen Cache-Cluster kopiert. Die Dauer dieses Prozesses ist abhängig vom Knotentyp und der Datenmenge im Cache-Cluster.
3. Lese- und Schreibvorgänge werden jetzt unter Verwendung des neuen Cache-Clusters ausgeführt. Da die Endpunkte des neuen Cache-Clusters mit denen des alten Cache-Clusters übereinstimmen, müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Sie werden eine kurze Unterbrechung (einige Sekunden) der Lese- und Schreibvorgänge auf dem Primärknoten feststellen, während der DNS Eintrag aktualisiert wird.
4. ElastiCache löscht den alten Cache-Cluster. Sie werden eine kurze Unterbrechung (einige Sekunden) der Lese- und Schreiboperationen des alten Knotens feststellen, da die Verbindungen zum alten Knoten getrennt werden.

### Note

Bei Clustern, auf denen der R6gd-Knotentyp ausgeführt wird, können Sie nur auf Knotengrößen innerhalb der R6gd-Knotenfamilie skalieren.

Wie in der folgenden Tabelle dargestellt, ist Ihr Valkey- oder OSS Redis-Scale-up-Vorgang blockiert, wenn Sie für das nächste Wartungsfenster ein Engine-Upgrade geplant haben. Weitere Informationen zu Wartungsfenstern finden Sie unter [Verwaltung der ElastiCache Cluster-Wartung](#).

### Blockierte Valkey- oder Redis-Operationen OSS

Ausstehende Operationen	Blockierte Operationen
Aufwärtsskalierung	Unmittelbares Engine-Upgrade
Engine-Upgrade	Unmittelbares Aufwärtsskalieren
Aufwärtsskalierung und Engine-Upgrade	Unmittelbares Aufwärtsskalieren

Ausstehende Operationen	Blockierte Operationen
	Unmittelbares Engine-Upgrade

Wenn Sie durch ausstehende Operationen blockiert werden, haben Sie folgende Möglichkeiten.

- Planen Sie Ihren Valkey- oder OSS Redis-Scale-up-Vorgang für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Sofort anwenden deaktivieren (CLI verwenden: `verwenden:--no-apply-immediately`). API `ApplyImmediately=false`
- Warten Sie bis zum nächsten Wartungsfenster (oder danach), um Ihren Valkey- oder OSS Redis-Scale-Up-Vorgang durchzuführen.
- Fügen Sie das Valkey- oder OSS Redis-Engine-Upgrade zu dieser Cache-Cluster-Änderung hinzu, wobei das Kontrollkästchen Sofort anwenden aktiviert ist (use: `--apply-immediately`, CLI API use: `use:ApplyImmediately=true`). Dadurch wird die Skalierung freigegeben, da das Engine-Upgrade sofort ausgeführt wird.

Sie können einen Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit einem einzigen Knoten mithilfe der ElastiCache Konsole, der oder oder hochskalieren. AWS CLI ElastiCache API

#### Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder OSS Redis-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

## Skalierung von Clustern mit einem Knoten für Valkey oder Redis OSS (Clustermodus deaktiviert) (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie einen Valkey- oder OSS Redis-Cluster mit einem Knoten mithilfe der Management Console skalieren. ElastiCache Während dieses Vorgangs bearbeitet Ihr Valkey- oder OSS Redis-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

So skalieren Sie einen Valkey- oder OSS Redis-Cluster mit einem Knoten (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey- oder OSS Redis-Cluster aus.
3. Wählen Sie aus der Clusterliste den Cluster aus, den Sie skalieren möchten (auf ihm muss die Valkey- oder Redis-Engine ausgeführt werden, nicht die geclusterte Valkey- oder OSS Redis-Engine). OSS
4. Wählen Sie Ändern aus.
5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
  - a. Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten.
  - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
6. Wenn Sie die Skalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately. Wenn Sie das Kontrollkästchen für Apply immediately nicht aktivieren, wird die Skalierung während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus.

Wenn Sie im vorherigen Schritt Apply immediately aktiviert haben, ändert sich der Cluster-Status zu `modifying`. Wenn der Status zu `available` wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

## Skalierung von Valkey- oder Redis-Cache-Clustern mit einem Knoten () OSS AWS CLI

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten mithilfe von skalieren. AWS CLI Während dieses Vorgangs bearbeitet Ihr Valkey- oder OSS Redis-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten hochzuskalieren (AWS CLI)

1. Ermitteln Sie die Knotentypen, auf die Sie skalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

- `--cache-cluster-id`

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
 --cache-cluster-id my-cache-cluster-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
 --cache-cluster-id my-cache-cluster-id
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

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

```
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-Änderungen](#) in der AWS CLI Referenz.

2. Ändern Sie Ihren vorhandenen Cache-Cluster, indem Sie den zu skalierenden Cache-Cluster und den neuen, größeren Knotentyp angeben. Verwenden Sie dazu den AWS CLI `modify-cache-cluster` Befehl und die folgenden Parameter.
  - `--cache-cluster-id` – Der Name des aufzuskalierenden Cache-Clusters.
  - `--cache-node-type` Der neue Knotentyp, auf den der Cache-Cluster skaliert werden soll. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
  - `--cache-parameter-group-name` – [Optional) Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
  - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

```
{
 "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"
 }
}
```

Weitere Informationen finden Sie unter [modify-cache-cluster](#) in der AWS CLI Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des neuen Cache-Clusters mithilfe des AWS CLI `describe-cache-clusters` Befehls mit dem folgenden Parameter. Wenn der Status zu Verfügbar wechselt, können Sie den neuen, größeren Cache-Cluster verwenden.
  - `--cache-cluster-id`— Der Name Ihres Valkey- oder OSS Redis-Cache-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cache-Cluster zu beschreiben, statt alle Cache-Cluster anzugeben.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Weitere Informationen finden Sie unter [describe-cache-clusters](#) in der Referenz.AWS CLI

Skalierung von Valkey- oder OSS Redis-Cache-Clustern mit einem Knoten () ElastiCache API

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten mithilfe von skalieren. ElastiCache API Während dieses Vorgangs bearbeitet Ihr Valkey- oder OSS Redis-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten hochzuskalieren () ElastiCache API

1. Ermitteln Sie die Knotentypen, auf die Sie skalieren können, indem Sie die ElastiCache API `ListAllowedNodeTypeModifications` Aktion mit dem folgenden Parameter ausführen.
  - `CacheClusterId`— Der Name des Valkey- oder OSS Redis-Cache-Clusters mit einem Knoten, den Sie hochskalieren möchten.

```
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>
```

Weitere Informationen finden Sie unter [ListAllowedNodeTypeModifications](#) in der ElastiCache APIAmazon-Referenz.

2. Ändern Sie Ihren vorhandenen Cache-Cluster, indem Sie den zu skalierenden Cache-Cluster und den neuen, größeren Knotentyp angeben. Verwenden Sie dazu die `ModifyCacheCluster` ElastiCache API Aktion und die folgenden Parameter.
  - `CacheClusterId` – Der Name des aufzuskalierenden Cache-Clusters.

- **CacheNodeType** Der neue, größere Knotentyp, auf den der Cache-Cluster skaliert werden soll. Dieser Wert muss einer der Knotentypen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
- **CacheParameterGroupName** – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
- **ApplyImmediately** Legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie `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>
```

Weitere Informationen finden Sie unter [ModifyCacheCluster](#) in der `ElastiCache API` Amazon-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überprüfen Sie den Status des neuen Cache-Clusters mithilfe der `ElastiCache API DescribeCacheClusters` Aktion mit dem folgenden Parameter. Wenn der Status zu `Verfügbar` wechselt, können Sie den neuen, größeren Cache-Cluster verwenden.
  - **CacheClusterId**— Der Name Ihres Valkey- oder OSS Redis-Cache-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cache-Cluster zu beschreiben, statt alle Cache-Cluster anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [DescribeCacheClusters](#) in der ElastiCache API Amazon-Referenz.

## Herunterskalierung von Valkey- oder Redis-Clustern mit einem Knoten OSS

In den folgenden Abschnitten erfahren Sie, wie Sie einen Valkey- oder OSS Redis-Cluster mit einem Knoten auf einen kleineren Knotentyp herunterskalieren können. Es ist wichtig für den langfristigen Erfolg Ihres neuen Valkey- oder Redis-Clusters sicherzustellen, dass der neue, kleinere Knotentyp groß genug ist, um alle Daten und den Valkey- oder OSS Redis-Overhead aufzunehmen. OSS Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#).

### Note

Bei Clustern, auf denen der R6gd-Knotentyp ausgeführt wird, können Sie nur auf Knotengrößen innerhalb der R6gd-Knotenfamilie skalieren.

## Themen

- [Herunterskalierung eines Valkey- oder Redis-Clusters mit einem Knoten \(Konsole\) OSS](#)
- [Herunterskalierung der Valkey- oder Redis-Cache-Cluster mit einem Knoten \(\) OSS AWS CLI](#)
- [Herunterskalierung der Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten \(\) ElastiCache API](#)

## Herunterskalierung eines Valkey- oder Redis-Clusters mit einem Knoten (Konsole) OSS

Das folgende Verfahren führt Sie durch die Skalierung Ihres Valkey- oder OSS Redis-Clusters mit einem Knoten auf einen kleineren Knotentyp mithilfe der Konsole. ElastiCache

### Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder OSS Redis-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

Um Ihren Valkey- oder OSS Redis-Cluster (Konsole) mit einem Knoten zu verkleinern

1. Vergewissern Sie sich, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht.
2. Wenn Ihre Parametergruppe Speicher für Valkey- oder OSS Redis-Overhead reserviert, stellen Sie sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, um die richtige Speichermenge für Ihren neuen Knotentyp bereitzustellen. `reserved-memory`

Alternativ können Sie eine benutzerdefinierte Parametergruppe für die Verwendung von `reserved-memory-percent` anpassen. Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

3. Melden Sie sich bei an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
4. Wählen Sie aus der Liste der Cluster den Cluster aus, den Sie nach unten skalieren möchten. Auf diesem Cluster muss die Valkey- oder OSS Redis-Engine ausgeführt werden und nicht die geclusterte Valkey- oder Redis-Engine. OSS
5. Wählen Sie Ändern aus.
6. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
  - a. Wählen Sie in der Liste Node type (Knotentyp) den Knotentyp aus, auf den Sie herunterskalieren möchten.
  - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
7. Wenn Sie die Skalierung nach unten sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately (Sofort anwenden). Wenn Sie das Kontrollkästchen für Apply immediately (Sofort anwenden) nicht aktivieren, wird die Skalierung nach unten während des nächsten Wartungsfensters des Clusters durchgeführt.
8. Wählen Sie Ändern aus.
9. Wenn der Status des Clusters von `modifying` zu `available` wechselt, wurde die Skalierung auf den neuen Knotentyp erfolgreich abgeschlossen. Die Endpunkte müssen nicht in der Anwendung aktualisiert werden.

## Herunterskalierung der Valkey- oder Redis-Cache-Cluster mit einem Knoten () OSS AWS CLI

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten mithilfe von herunterskalieren. AWS CLI

So verkleinern Sie einen Valkey- oder Redis-Cache-Cluster mit einem Knoten () OSS AWS CLI

1. Ermitteln Sie die Knotentypen, auf die Sie herunterskalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

- `--cache-cluster-id`

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
 --cache-cluster-id my-cache-cluster-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
 --cache-cluster-id my-cache-cluster-id
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

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

Weitere Informationen finden Sie unter [list-allowed-node-type-Änderungen](#) in der AWS CLI Referenz.

2. Ändern Sie Ihren vorhandenen Cache-Cluster, indem Sie den Cache-Cluster, der herunterskaliert werden soll, und den neuen, kleineren Knotentyp angeben. Verwenden Sie dazu den AWS CLI `modify-cache-cluster` Befehl und die folgenden Parameter.

- `--cache-cluster-id` – Der Name des herunter zu skalierenden Cache-Clusters.
- `--cache-node-type` Der neue Knotentyp, auf den der Cache-Cluster skaliert werden soll. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
- `--cache-parameter-group-name` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
- `--apply-immediately` – Bewirkt, dass der Verkleinerungsprozess sofort angewendet wird. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

```
{
 "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"
 }
}
```

Weitere Informationen finden Sie unter [modify-cache-cluster](#) in der AWS CLI Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des neuen Cache-Clusters mithilfe des AWS CLI `describe-cache-clusters` Befehls mit dem folgenden

Parameter. Wenn der Status zu Verfügbar wechselt, können Sie den neuen, größeren Cache-Cluster verwenden.

- `--cache-cluster-id`— Der Name Ihres Valkey- oder OSS Redis-Cache-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cache-Cluster zu beschreiben, statt alle Cache-Cluster anzugeben.

```
aws elasticache describe-cache-clusters --cache-cluster-id my-redis-cache-cluster
```

Weitere Informationen finden Sie unter [describe-cache-clusters](#) in der Referenz.AWS CLI

## Herunterskalierung der Valkey- oder OSS Redis-Cache-Cluster mit einem Knoten () ElastiCache API

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder Redis-Cache-Cluster mit einem einzelnen Knoten mithilfe von hochskalieren und herunterskalieren. OSS ElastiCache API

So verkleinern Sie einen Valkey- oder Redis-Cache-Cluster mit einem Knoten () OSS ElastiCache API

1. Ermitteln Sie die Knotentypen, auf die Sie herunterskalieren können, indem Sie die ElastiCache API `ListAllowedNodeTypeModifications` Aktion mit dem folgenden Parameter ausführen.
  - `CacheClusterId`— Der Name des Valkey- oder OSS Redis-Cache-Clusters mit einem Knoten, den Sie herunterskalieren möchten.

```
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>
```

Weitere Informationen finden Sie unter [ListAllowedNodeTypeModifications](#) in der ElastiCache API Amazon-Referenz.

2. Ändern Sie Ihren vorhandenen Cache-Cluster, indem Sie den zu skalierenden Cache-Cluster und den neuen, größeren Knotentyp angeben. Verwenden Sie dazu die `ModifyCacheCluster` ElastiCache API Aktion und die folgenden Parameter.
  - `CacheClusterId` – Der Name des herunter zu skalierenden Cache-Clusters.
  - `CacheNodeType` – Der neue Knotentyp, auf den der Cache-Cluster herunterskaliert werden soll. Dieser Wert muss einer der Knotentypen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
  - `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
  - `ApplyImmediately` – Legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie `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>
```

Weitere Informationen finden Sie unter [ModifyCacheCluster](#) in der ElastiCache API Amazon-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überprüfen Sie den Status des neuen Cache-Clusters mithilfe der ElastiCache API `DescribeCacheClusters` Aktion mit dem folgenden Parameter. Wenn der Status zu verfügbar wechselt, können Sie den neuen, kleineren Cache-Cluster verwenden.

- `CacheClusterId`— Der Name Ihres Valkey- oder OSS Redis-Cache-Clusters mit einem Knoten. Verwenden Sie diesen Parameter, um einen bestimmten Cache-Cluster zu beschreiben, statt alle Cache-Cluster anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [DescribeCacheClusters](#) in der ElastiCache API Amazon-Referenz.

## Skalierung von Replikatknoten für Valkey oder Redis OSS (Cluster-Modus deaktiviert)

Ein Valkey- oder OSS Redis-Cluster mit Replikatknoten (im FeldAPI/als Replikationsgruppe bezeichnetCLI) bietet Hochverfügbarkeit durch Replikation, bei der Multi-AZ mit automatischem Failover aktiviert ist. Ein Cluster mit Replikatknoten ist eine logische Sammlung von bis zu sechs Valkey- oder OSS Redis-Knoten, wobei ein Knoten, der primäre, sowohl Lese- als auch Schreibenanforderungen bearbeiten kann. Alle anderen Knoten im Cluster sind reine Read Replicas des primären Clusters. Daten, die im primären Cluster geschrieben werden, werden asynchron auf die Read Replicas im Cluster repliziert. Da Valkey oder Redis OSS (Clustermodus deaktiviert) die Partitionierung Ihrer Daten auf mehrere Cluster nicht unterstützen, enthält jeder Knoten in einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) den gesamten Cache-Datensatz. Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert) unterstützen die Partitionierung Ihrer Daten auf bis zu 500 Shards.

Um die Datenkapazität Ihres Clusters zu ändern, müssen Sie es auf einen größeren Knotentyp hoch- oder auf einen kleineren Knotentyp herabskalieren.

Um die Lesekapazität Ihres Clusters zu ändern, fügen Sie bis auf maximal 5 weitere Lesereplikate hinzu oder entfernen Sie Lesereplikate.

Der ElastiCache Skalierungsprozess ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich beizubehalten, und erfordert eine erfolgreiche Valkey- oder Redis-Replikation. OSS Für Valkey- oder OSS Redis-Cluster mit Replikaten empfehlen wir, Valkey oder Redis ausreichend Speicher zur Verfügung zu stellen. OSS

### Themen

- [Skalierung von Valkey- oder Redis-Clustern mit Replikaten OSS](#)
- [Herunterskalierung von Valkey- oder OSS Redis-Clustern mit Replikaten](#)
- [Erhöhen der Lesekapazität](#)
- [Verringern der Lesekapazität](#)

### Verwandte Themen

- [Hohe Verfügbarkeit mit Replikationsgruppen](#)
- [Replikation: Valkey- und OSS Redis-Clustermodus deaktiviert oder aktiviert](#)
- [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#)

- [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#)

## Themen

- [Skalierung von Valkey- oder Redis-Clustern mit Replikaten OSS](#)
- [Herunterskalierung von Valkey- oder OSS Redis-Clustern mit Replikaten](#)
- [Erhöhen der Lesekapazität](#)
- [Verringern der Lesekapazität](#)

## Skalierung von Valkey- oder Redis-Clustern mit Replikaten OSS

Amazon ElastiCache bietet Konsole und API Unterstützung für die Skalierung Ihrer Valkey- oder Redis-Replikationsgruppe OSS (Cluster-Modus deaktiviert). CLI

Wenn der Scale-Up-Prozess eingeleitet wird, ElastiCache geht Folgendes vor:

1. Es wird eine Replikationsgruppe unter Verwendung des neuen Knotentyps gestartet.
2. Alle Daten aus dem derzeitigen primären Knoten werden in den neuen primären Knoten kopiert.
3. Die neuen Read Replicas werden mit dem neuen primären Knoten synchronisiert.
4. Aktualisiert die DNS Einträge, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei OSS Redis-Version 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der Eintrag aktualisiert wird. DNS
5. Löscht die alten Knoten (CLI/API: Replikationsgruppe). Sie werden eine kurze Unterbrechung (ein paar Sekunden) der Lese- und Schreiboperationen der alten Knoten feststellen, da die Verbindungen zu den alten Knoten getrennt werden.

Die Dauer dieses Prozess ist abhängig vom Knotentyp und der Datenmenge im Cluster.

Wie in der folgenden Tabelle dargestellt, ist Ihr Valkey- oder OSS Redis-Scale-up-Vorgang blockiert, wenn Sie ein Engine-Upgrade für das nächste Wartungsfenster des Clusters geplant haben.

### Blockierte Valkey- oder Redis-Operationen OSS

Ausstehende Operationen	Blockierte Operationen
Aufwärtsskalierung	Unmittelbares Engine-Upgrade
Engine-Upgrade	Unmittelbares Aufwärtsskalieren
Aufwärtsskalierung und Engine-Upgrade	Unmittelbares Aufwärtsskalieren Unmittelbares Engine-Upgrade

Wenn Sie durch ausstehende Operationen blockiert werden, haben Sie folgende Möglichkeiten.

- Planen Sie Ihren Valkey- oder OSS Redis-Scale-up-Vorgang für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Sofort anwenden deaktivieren (CLI verwenden: `verwenden:--no-apply-immediately`). API `ApplyImmediately=false`
- Warten Sie bis zum nächsten Wartungsfenster (oder danach), um Ihren Valkey- oder Redis-Scale-up-Vorgang durchzuführen. OSS
- Fügen Sie das Valkey- oder OSS Redis-Engine-Upgrade zu dieser Cache-Cluster-Änderung hinzu, wobei das Kontrollkästchen Sofort anwenden aktiviert ist (CLI use: `use:--apply-immediately`). API `ApplyImmediately=true` Dadurch wird die Skalierung freigegeben, da das Engine-Upgrade sofort ausgeführt wird.

In den folgenden Abschnitten wird beschrieben, wie Sie Ihren Valkey- oder OSS Redis-Cluster mit Replikaten mithilfe der ElastiCache Konsole, der und der skalieren. AWS CLI ElastiCache API

#### Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder OSS Redis-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

## Skalieren eines Valkey- oder OSS Redis-Clusters mit Replikaten (Konsole)

Wie lange die Aufwärtsskalierung auf einen größeren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cluster ab.

Der folgende Prozess skaliert Ihren Cluster mit Replikaten mithilfe der Konsole vom aktuellen Knotentyp auf einen neuen, größeren Knotentyp. ElastiCache Während dieses Vorgangs kann es zu einer kurzen Unterbrechung der Lese- und Schreibvorgänge für andere Versionen vom primären Knoten kommen, während der DNS Eintrag aktualisiert wird. Bei Knoten, die auf Versionen 5.0.6 und höher ausgeführt werden, kann es zu einer Ausfallzeit von weniger als 1 Sekunde und bei älteren Versionen zu einigen Sekunden kommen.

## So skalieren Sie den Valkey- oder OSS Redis-Cluster mit Replikaten (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus
3. Wählen Sie aus der Liste der Cluster den Cluster aus, den Sie skalieren möchten. Auf diesem Cluster muss die Valkey- oder Redis-Engine ausgeführt werden und nicht die geclusterte Valkey- oder OSS Redis-Engine. OSS
4. Wählen Sie Ändern aus.
5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
  - a. Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten. Beachten Sie, dass nicht alle Knotentypen für das Herunterskalieren zur Verfügung stehen.
  - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
6. Wenn Sie die Skalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately. Wenn Sie das Kontrollkästchen für Apply immediately nicht aktivieren, wird die Skalierung während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus.
8. Wenn der Status des Clusters von `modifying` zu `available` wechselt, wurde die Skalierung auf den neuen Knotentyp erfolgreich abgeschlossen. Die Endpunkte müssen nicht in der Anwendung aktualisiert werden.

## Skalierung einer Valkey- oder Redis-Replikationsgruppe () OSS AWS CLI

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der AWS CLI vom derzeitigen Knotentyp auf einen neuen, größeren Knotentyp skaliert. Während dieses Vorgangs werden die DNS Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. In Version 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der Eintrag aktualisiert wird. DNS

Wie lange die Aufwärtsskalierung auf einen größeren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

So skalieren Sie eine Valkey- oder OSS Redis-Replikationsgruppe ( )AWS CLI

1. Ermitteln Sie, auf welche Knotentypen Sie skalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.
  - `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-repl-group
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
 --replication-group-id my-repl-group
```

Die Ausgabe dieser Operation sieht ungefähr so aus (JSONFormat).

```
{
 "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"
]
}
```

```
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-Änderungen](#) in der AWS CLI Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe des AWS CLI `modify-replication-group` Befehls mit den folgenden Parametern auf den neuen Knotentyp.
  - `--replication-group-id` der Name der Replikationsgruppe.
  - `--cache-node-type` der neue, größere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Dieser Wert muss einer der Instanztypen sein, die vom `list-allowed-node-type-modifications` Befehl im vorherigen Schritt zurückgegeben wurden.
  - `--cache-parameter-group-name` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
  - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `--no-apply-immediately`.

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe dieses Befehls sieht ungefähr so aus (JSONFormat).

```
{
 "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": {}
}
}
```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI Referenz.

3. Wenn Sie den `--apply-immediately` Parameter verwendet haben, überwachen Sie den Status der Replikationsgruppe mithilfe des AWS CLI `describe-replication-group` Befehls mit dem folgenden Parameter. Solange sich der Status noch in Bearbeitung befindet, kann es zu einer Ausfallzeit von weniger als 1 Sekunde für Knoten kommen, auf denen die Versionen 5.0.6 und höher ausgeführt werden, und es kommt zu einer kurzen Unterbrechung der Lese- und Schreibvorgänge für ältere Versionen auf dem primären Knoten, während der DNS Eintrag aktualisiert wird.
- `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```
aws elasticache describe-replication-groups \
 --replication-group-id my-replication-group
```

Für Windows:

```
aws elasticache describe-replication-groups ^
 --replication-group-id my-replication-group
```

Weitere Informationen finden Sie [describe-replication-groups](#) in der AWS CLI Referenz.

## Skalieren einer Valkey- oder OSS Redis-Replikationsgruppe () ElastiCache API

Der folgende Prozess skaliert Ihre Replikationsgruppe von ihrem aktuellen Knotentyp auf einen neuen, größeren Knotentyp mithilfe von ElastiCache API. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Version Redis OSS 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der Eintrag aktualisiert wird. DNS

Wie lange die Aufwärtsskalierung auf einen größeren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

So skalieren Sie eine Valkey- oder OSS Redis-Replikationsgruppe () ElastiCache API

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache API `ListAllowedNodeTypeModifications` Aktion mit dem folgenden Parameter skalieren können.

- `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [ListAllowedNodeTypeModifications](#) in der ElastiCache API Amazon-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyRedplicationGroup` ElastiCache API Aktion und mit den folgenden Parametern auf den neuen Knotentyp.

- `ReplicationGroupId` der Name der Replikationsgruppe.
- `CacheNodeType` der neue, größere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Dieser Wert muss einer der Instanztypen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
- `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.

- `ApplyImmediately` legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `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>
```

Weitere Informationen finden Sie unter [ModifyReplicationGroup](#) in der [ElastiCache API Amazon-Referenz](#).

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überwachen Sie den Status der Replikationsgruppe mithilfe der [ElastiCache API DescribeReplicationGroups](#) Aktion mit den folgenden Parametern. Wenn der Status von `modifying` zu `available` wechselt, können Sie die neue, aufskalierte Replikationsgruppe verwenden.
  - `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [DescribeReplicationGroups](#) in der ElastiCache API Amazon-Referenz.

## Herunterskalierung von Valkey- oder OSS Redis-Clustern mit Replikaten

In den folgenden Abschnitten erfahren Sie, wie Sie einen Valkey- oder Redis-Cache-Cluster OSS (Clustermodus deaktiviert) mit Replikatknoten auf einen kleineren Knotentyp herunterskalieren. Es ist für den Erfolg wichtig sicherzustellen, dass der neue, kleinere Knotentyp für die Datenmenge und den Overhead ausreicht. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#).

### Note

Bei Clustern, auf denen der R6gd-Knotentyp ausgeführt wird, können Sie nur auf Knotengrößen innerhalb der R6gd-Knotenfamilie skalieren.

### Important

Wenn Ihre Parametergruppe Speicher für Valkey- oder OSS Redis-Overhead reserviert, stellen Sie vor Beginn der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die richtige Speichermenge für Ihren neuen Knotentyp reserviert. `reserved-memory` Alternativ können Sie eine benutzerdefinierte Parametergruppe anpassen, die `reserved-memory-percent` verwendet, und diese Parametergruppe für Ihren neuen Cluster verwenden.

Wenn Sie `reserved-memory-percent` verwenden, ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

## Herunterskalieren eines Valkey- oder OSS Redis-Clusters mit Replikaten (Konsole)

Der folgende Prozess skaliert Ihren Valkey- oder OSS Redis-Cluster mit Replikatknoten mithilfe der Konsole auf einen kleineren Knotentyp. ElastiCache

### So verkleinern Sie einen Valkey- oder OSS Redis-Cluster mit Replikatknoten (Konsole)

1. Vergewissern Sie sich, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht.
2. Wenn Ihre Parametergruppe Speicher für Valkey- oder OSS Redis-Overhead reserviert, stellen Sie sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, um die richtige Speichermenge für Ihren neuen Knotentyp bereitzustellen. `reserved-memory`

Alternativ können Sie eine benutzerdefinierte Parametergruppe für die Verwendung von `reserved-memory-percent` anpassen. Weitere Informationen finden Sie unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

3. Melden Sie sich bei an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
4. Wählen Sie aus der Liste der Cluster den Cluster aus, den Sie nach unten skalieren möchten. Auf diesem Cluster muss die Valkey- oder OSS Redis-Engine ausgeführt werden und nicht die geclusterte Valkey- oder Redis-Engine. OSS
5. Wählen Sie Ändern aus.
6. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
  - a. Wählen Sie in der Liste Node type (Knotentyp) den Knotentyp aus, auf den Sie herunterskalieren möchten.
  - b. Wenn Sie zur Speicherverwaltung `reserved-memory` verwenden, wählen Sie in der Liste Parameter Group die benutzerdefinierte Parametergruppe aus, die die korrekte Speichermenge für den neuen Knotentyp reserviert.
7. Wenn Sie die Skalierung nach unten sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately (Sofort anwenden). Wenn Sie das Kontrollkästchen für Apply immediately (Sofort anwenden) nicht aktivieren, wird die Skalierung nach unten während des nächsten Wartungsfensters des Clusters durchgeführt.
8. Wählen Sie Ändern aus.
9. Wenn der Status des Clusters von `modifying` zu `available` wechselt, wurde die Skalierung auf den neuen Knotentyp erfolgreich abgeschlossen. Die Endpunkte müssen nicht in der Anwendung aktualisiert werden.

## Herunterskalieren einer Valkey- oder Redis-Replikationsgruppe () OSS AWS CLI

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der AWS CLI vom derzeitigen Knotentyp auf einen neuen, kleineren Knotentyp skaliert. Während dieses Vorgangs werden die DNS Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. In Version 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der Eintrag aktualisiert wird. DNS

Lesevorgänge von den Read Replica-Cache-Clustern werden jedoch fortgesetzt.

Wie lange die Abwärtsskalierung auf einen kleineren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

Um eine Valkey- oder OSS Redis-Replikationsgruppe herunterzuskalieren (AWS CLI)

1. Ermitteln Sie, auf welche Knotentypen Sie herunterskalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.
  - `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-repl-group
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
 --replication-group-id my-repl-group
```

Die Ausgabe dieser Operation sieht ungefähr so aus (JSONFormat).

```
{
 "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"
]
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-Änderungen](#) in der AWS CLI Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe des AWS CLI `modify-replication-group` Befehls mit den folgenden Parametern auf den neuen Knotentyp.

- `--replication-group-id` der Name der Replikationsgruppe.
- `--cache-node-type` Der neue, kleinere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Dieser Wert muss einer der Instanztypen sein, die vom `list-allowed-node-type-modifications` Befehl im vorherigen Schritt zurückgegeben wurden.
- `--cache-parameter-group-name` – [Optional) Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
- `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `--no-apply-immediately`.

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe dieses Befehls sieht ungefähr so aus (JSONFormat).

```
{
 "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": {}
}
}
```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI Referenz.

3. Wenn Sie den `--apply-immediately` Parameter verwendet haben, überwachen Sie den Status der Replikationsgruppe mithilfe des AWS CLI `describe-replication-group` Befehls mit dem folgenden Parameter. Wenn der Status von In Bearbeitung zu Verfügbar wechselt, können Sie die neue, abwärts skalierte Replikationsgruppe verwenden.
- `--replication-group-id` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

Für Linux, macOS oder Unix:

```
aws elasticache describe-replication-group \
 --replication-group-id my-replication-group
```

Für Windows:

```
aws elasticache describe-replication-groups ^
 --replication-group-id my-replication-group
```

Weitere Informationen finden Sie [describe-replication-groups](#) in der AWS CLI Referenz.

Eine Valkey- oder OSS Redis-Replikationsgruppe herunterskalieren () ElastiCache API

Der folgende Prozess skaliert Ihre Replikationsgruppe von ihrem aktuellen Knotentyp auf einen neuen, kleineren Knotentyp mithilfe von. ElastiCache API Während dieses Vorgangs werden die DNS Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie

die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher oder Redis OSS 5.0.5 und höher können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei OSS Redis-Version 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der Eintrag aktualisiert wird. DNS Lesevorgänge von den Read Replica-Cache-Clustern werden jedoch fortgesetzt.

Wie lange die Abwärtsskalierung auf einen kleineren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

Um eine Valkey- oder OSS Redis-Replikationsgruppe herunterzuskalieren () ElastiCache API

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache API `ListAllowedNodeTypeModifications` Aktion mit dem folgenden Parameter herunterskalieren können.
  - `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [ListAllowedNodeTypeModifications](#) in der ElastiCache API Amazon-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyRedplicationGroup` ElastiCache API Aktion und mit den folgenden Parametern auf den neuen Knotentyp.
  - `ReplicationGroupId` der Name der Replikationsgruppe.
  - `CacheNodeType` Der neue, kleinere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Dieser Wert muss einer der Instanztypen sein, die von der

ListAllowedNodeTypeModifications Aktion im vorherigen Schritt zurückgegeben wurden.

- CacheParameterGroupName – [Optional) Verwenden Sie diesen Parameter, wenn Sie reserved-memory verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie reserved-memory-percent verwenden, können Sie diesen Parameter weglassen.
- ApplyImmediatelyLegen Sie true fest, um die Skalierung sofort auszuführen. Wenn Sie die Abwärtsskalierung für das nächste Wartungsfenster planen möchten, verwenden Sie 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>
```

Weitere Informationen finden Sie unter [ModifyReplicationGroup](#) in der ElastiCache API Amazon-Referenz.

3. Falls Sie dies verwendet haben ApplyImmediately=true, überwachen Sie den Status der Replikationsgruppe mithilfe der ElastiCache API DescribeReplicationGroups Aktion mit den folgenden Parametern. Wenn der Status von In Bearbeitung zu Verfügbar wechselt, können Sie die neue, abwärts skalierte Replikationsgruppe verwenden.
  - ReplicationGroupId der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [DescribeReplicationGroups](#) in der ElastiCache API Amazon-Referenz.

## Erhöhen der Lesekapazität

Um die Lesekapazität zu erhöhen, fügen Sie Read Replicas (bis zu maximal fünf) zu Ihrer Valkey- oder OSS Redis-Replikationsgruppe hinzu.

Sie können die Lesekapazität Ihres Valkey- oder OSS Redis-Clusters mithilfe der ElastiCache Konsole, der oder der skalieren. AWS CLI ElastiCache API Weitere Informationen finden Sie unter [Hinzufügen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

## Verringern der Lesekapazität

Um die Lesekapazität zu verringern, löschen Sie eine oder mehrere Read Replicas aus Ihrem Valkey- oder OSS Redis-Cluster mit Replikaten (im/als Replikationsgruppe bezeichnet). API CLI Wenn es sich bei dem Cluster um Multi-AZ mit aktiviertem automatischem Failover handelt, können Sie das letzte Read Replica nicht löschen, ohne zuvor Multi-AZ zu deaktivieren. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).

Weitere Informationen finden Sie unter [Löschen einer Read Replica für Valkey oder Redis OSS \(Cluster-Modus deaktiviert\)](#).

## Skalierung von Clustern in Valkey oder Redis OSS (Clustermodus aktiviert)

Wenn sich die Nachfrage nach Ihren Clustern ändert, können Sie entscheiden, die Leistung zu verbessern oder die Kosten zu senken, indem Sie die Anzahl der Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) ändern. Wir empfehlen, dazu die horizontale Online-Skalierung zu verwenden, da Ihr Cluster während des Skalierungsprozesses weiterhin Anforderungen bedienen kann.

Zu den Bedingungen, unter denen Sie den Cluster möglicherweise neu skalieren, gehören folgende:

- Speicherbelastung

Wenn die Knoten in Ihrem Cluster einer hohen Speicherbelastung ausgesetzt sind, können Sie sich für eine Skalierung entscheiden, sodass Sie über mehr Ressourcen verfügen, um Daten besser speichern und Anforderungen verarbeiten zu können.

Sie können feststellen, ob Ihre Knoten unter Speicherauslastung stehen, indem Sie die folgenden Messwerte überwachen: `FreeableMemory`, und `SwapUsage`. `BytesUseForCache`

- CPU oder Netzwerkengpass:

Wenn Latenz-/Durchsatzprobleme Ihren Cluster belasten, müssen Sie möglicherweise zur Lösung der Probleme eine Skalierung durchführen.

Sie können Ihre Latenz und Ihren Durchsatz überwachen, indem Sie die folgenden Messwerte überwachen: `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOutCurrConnections`, und `NewConnections`

- Ihr Cluster ist übermäßig skaliert:

Der aktuelle Bedarf an Ihrem Cluster ist so hoch, dass eine Skalierung die Leistung nicht beeinträchtigt und Ihre Kosten reduziert.

Sie können die Nutzung Ihres Clusters überwachen, um anhand der folgenden Metriken festzustellen, ob Sie sicher skalieren können: `FreeableMemorySwapUsageBytesUseForCache`, `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, `CurrConnections`, und `NewConnections`.

### Leistungsbeeinträchtigung durch Skalierung

Wenn Sie den Offline-Prozess skalieren, ist Ihr Cluster für einen erheblichen Teil des Prozesses offline und kann daher keine Anforderungen erfüllen. Wenn Sie mithilfe der Onlinemethode

skalieren, da die Skalierung eine rechenintensive Operation ist, gibt es eine Leistungseinbuße. Ihr Cluster führt jedoch weiterhin während der Skalierungsoperation Anforderungen aus. Wie stark die Beeinträchtigung ist, hängt von Ihrer normalen CPU Auslastung und Ihren Daten ab.

Es gibt zwei Möglichkeiten, Ihren Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) zu skalieren: horizontale und vertikale Skalierung.

- Mit horizontaler Skalierung ändern Sie die Anzahl der Knotengruppen (Shards) in der Replikationsgruppe, indem Sie Knotengruppen hinzufügen oder entfernen. Online-Resharding ermöglicht Ein- und Ausskalieren, während der Cluster weiterhin alle eingehenden Anforderungen bedient.

Konfigurieren Sie die Slots in Ihrem neuen Cluster anders als im alten Cluster. Nur Offline-Methode.

- Vertikale Skalierung – Ändern Sie den Knotentyp, um die Größe des Clusters anzupassen. Vertikale Online-Skalierung ermöglicht Aufwärts- und Abwärtsskalieren, während der Cluster weiterhin alle eingehenden Anforderungen bedient.

Wenn Sie die Größe und Speicherkapazität des Clusters reduzieren, indem Sie entweder ein- oder herunterskalieren, stellen Sie sicher, dass die neue Konfiguration über ausreichend Speicher für Ihre Daten und Valkey- oder Redis-Overhead verfügt. OSS

Weitere Informationen finden Sie unter [Auswahl der Cache-Knotengröße](#).

## Inhalt

- [Offline-Resharding für Valkey oder OSS Redis \(Clustermodus aktiviert\)](#)
- [Online-Resharding für Valkey oder OSS Redis \(Clustermodus aktiviert\)](#)
  - [Hinzufügen von Shards mit Online-Resharding](#)
  - [Entfernen von Shards mit Online-Resharding](#)
    - [Entfernen von Shards \(Konsole\)](#)
    - [Entfernen von Shards \(AWS CLI\)](#)
    - [Shards werden entfernt \(\) ElastiCache API](#)
  - [Online-Shard-Rebalancing](#)
    - [Online-Shard-Rebalancing \(Console\)](#)
    - [Online-Shard-Rebalancing \(AWS CLI\)](#)
    - [Online-Shard-Rebalancing \(ElastiCache API\)](#)

- [Vertikales Online-Skalieren durch Ändern des Knotentyps](#)
  - [Online-aufwärtsskalieren](#)
    - [Skalierung von Valkey- oder OSS Redis-Cache-Clustern \(Konsole\)](#)
    - [Skalieren von Valkey- oder OSS Redis-Cache-Clustern \(\)AWS CLI](#)
    - [Skalierung von Valkey- oder OSS Redis-Cache-Clustern \(\) ElastiCache API](#)
  - [Online-abwärtsskalieren](#)
    - [Herunterskalierung der Valkey- oder OSS Redis-Cache-Cluster \(Konsole\)](#)
    - [Herunterskalieren von Valkey- oder OSS Redis-Cache-Clustern \(\)AWS CLI](#)
    - [Herunterskalieren der Valkey- oder OSS Redis-Cache-Cluster \(\) ElastiCache API](#)

Offline-Resharding für Valkey oder OSS Redis (Clustermodus aktiviert)

Der Hauptvorteil der Offline-Shard-Rekonfiguration besteht darin, dass Sie mehr als nur das Hinzufügen oder Entfernen von Shards aus Ihrer Replikationsgruppe tun können. Wenn Sie Resharding und Rebalancing offline durchführen, können Sie nicht nur die Anzahl der Shards in Ihrer Replikationsgruppe ändern, sondern auch Folgendes tun:

 Note

Offline-Resharding wird auf Valkey- oder Redis-Clustern mit aktiviertem Data Tiering nicht unterstützt. OSS Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

- Ändern Sie den Knotentyp Ihrer Replikationsgruppe.
- Geben Sie die Availability Zone für jeden Knoten in der Replikationsgruppe an.
- Upgrade auf eine neuere Engine-Version.
- Geben Sie die Anzahl der Replikatknoten in jeder Shard unabhängig voneinander an.
- Geben Sie den Keyspace für jede Shard an.

Der Hauptnachteil der Offline-Shard-Neukonfiguration besteht darin, dass Ihr Cluster offline ist und mit dem Wiederherstellungsabschnitt des Prozesses beginnt und so lange fortfährt, bis Sie die Endpunkte in Ihrer Anwendung aktualisieren. Die Dauer, in der Ihr Cluster offline ist, hängt von der Datenmenge in Ihrem Cluster ab.

Um Ihre Shards Valkey- oder Redis-Cluster (Clustermodus aktiviert) offline neu zu konfigurieren OSS

1. Erstellen Sie ein manuelles Backup Ihres vorhandenen Valkey- oder Redis-Clusters. OSS Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).
2. Erstellen Sie einen neuen Cluster, indem Sie sie aus der Sicherung wiederherstellen. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).
3. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

Online-Resharding für Valkey oder OSS Redis (Clustermodus aktiviert)

Mithilfe von Online-Resharding und Shard-Rebalancing mit ElastiCache Valkey 7.2 oder neuer oder Redis OSS Version 3.2.10 oder neuer können Sie Ihren Valkey- oder Redis-Cluster (Clustermodus aktiviert) dynamisch und ohne Ausfallzeiten skalieren. OSS Dieser Ansatz bedeutet, dass Ihr Cluster weiterhin Anfragen bearbeiten kann, selbst wenn die Skalierung oder das Rebalancing in Bearbeitung ist.

Sie haben die folgenden Möglichkeiten:

- Skalierung — Erhöhen Sie die Lese- und Schreibkapazität, indem Sie Ihrem Valkey- oder Redis-Cluster (Clustermodus aktiviert) (Replikationsgruppe) Shards (Knotengruppen) hinzufügen. OSS

Wenn Sie Ihrer Replikationsgruppe einen oder mehrere Shards hinzufügen, entspricht die Anzahl der Knoten in jedem neuen Shard der Anzahl der Knoten in der kleinsten der vorhandenen Shards.

- Skalierung — Reduzieren Sie die Lese- und Schreibkapazität und damit die Kosten, indem Sie Shards aus Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) entfernen.
- Rebalance — Verschieben Sie die Keyspaces zwischen den Shards in Ihrem Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert), sodass sie so gleichmäßig wie möglich auf die Shards verteilt sind.

Sie haben nicht die folgenden Möglichkeiten:

- Konfigurieren Sie Shards unabhängig:

Sie können den Keyspace für Shards nicht unabhängig angeben. Um dies zu tun, müssen Sie den Offline-Prozess verwenden.

Derzeit gelten die folgenden Einschränkungen für Online-Resharding und Rebalancing: ElastiCache

- Diese Prozesse erfordern Valkey 7.2 und neuer oder Redis OSS Engine-Version 3.2.10 oder neuer. Informationen zum Aktualisieren der Engine-Version finden Sie unter [Versionsverwaltung für ElastiCache](#).
- Es gibt Einschränkungen bei Slots oder Keyspaces und großen Elementen:

Wenn ein Schlüssel in einem Shard ein großes Element enthält, wird dieser Schlüssel beim horizontalen Skalieren oder Neuausrichten nicht in eine neue Shard migriert. Diese Funktionalität kann zu unsymmetrischen Shards führen.

Wenn einer der Schlüssel in einer Shard ein großes Element enthält (Elemente größer als 256 MB nach der Serialisierung), wird diese Shard beim Skalieren nicht gelöscht. Diese Funktionalität kann dazu führen, dass einige Shards nicht gelöscht werden.

- Beim Ausskalieren entspricht die Anzahl der Knoten in neuen Shards der Anzahl der Knoten in der kleinsten vorhandenen Shard.
- Beim Ausskalieren werden alle Tags, die allen vorhandenen Shards gemeinsam sind, in die neuen Shards kopiert.
- Beim Skalieren eines Global Data Store-Clusters ElastiCache werden Funktionen nicht automatisch von einem der vorhandenen Knoten auf die neuen Knoten repliziert. Wir empfehlen, Ihre Funktionen nach der Aufskalierung Ihres Clusters in den (die) neuen Shard(s) zu laden, sodass alle Shards dieselben Funktionen aufweisen.

#### Note

In ElastiCache Valkey 7.2 und höher und in Redis OSS Version 7 und höher: Wenn Sie Ihren Cluster skalieren, repliziert er ElastiCache automatisch die Funktionen, die in einem der vorhandenen Knoten geladen sind (zufällig ausgewählt), auf die neuen Knoten. Wenn Ihre Anwendung [Functions](#) verwendet, empfehlen wir, alle Ihre Funktionen vor dem Skalieren auf alle Shards zu laden, damit Ihr Cluster nicht mit unterschiedlichen Funktionsdefinitionen auf verschiedenen Shards endet.

Weitere Informationen finden Sie unter [Online-Größenanpassung von Clustern](#).

Sie können Ihre Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) horizontal skalieren oder neu ausbalancieren, indem Sie die AWS Management Console, und die verwenden. AWS CLI ElastiCache API

### Hinzufügen von Shards mit Online-Resharding

Sie können Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mithilfe von, oder Shards hinzufügen. AWS Management Console AWS CLI ElastiCache API Wenn Sie Shards zu einem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) hinzufügen, werden alle Tags auf den vorhandenen Shards auf die neuen Shards kopiert.

### Hinzufügen von Shards (Konsole)

Sie können den verwenden AWS Management Console , um Ihrem Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert) einen oder mehrere Shards hinzuzufügen. Das folgende Verfahren beschreibt den Prozess.

Um Shards zu Ihrem Valkey- oder OSS Redis-Cluster (Cluster-Modus aktiviert) hinzuzufügen

1. Öffnen Sie die Konsole unter ElastiCache . <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.
3. Suchen Sie den Namen des Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert), dem Sie Shards hinzufügen möchten, und wählen Sie ihn aus, nicht das Feld links neben dem Clusternamen.

#### Tip

Valkey oder Redis OSS (Clustermodus aktiviert) zeigen Clustered Valkey oder Clustered Redis in der Modus-Spalte an OSS

4. Wählen Sie Shard hinzufügen aus.
  - a. Wählen Sie für Anzahl der hinzuzufügenden Shardsdie Anzahl der Shards, die Sie diesem Cluster hinzufügen möchten.
  - b. Für Availability zone(s), entweder No preference oder Specify availability zones angeben.
  - c. Wenn Sie Availability Zones angebenfür jeden Knoten angeben möchten, wählen Sie für jeden Knoten in jeder Shard die Availability Zone des Knotens aus der Liste der Availability Zones.
  - d. Wählen Sie Hinzufügen aus.

## Shards hinzufügen (AWS CLI)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mit dem hinzufügen. AWS CLI

Verwenden Sie die folgenden Parameter mit `modify-replication-group-shard-configuration`.

### Parameter

- `--apply-immediately` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `--replication-group-id` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `--node-group-count` – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Hinzufügen von Shards muss der Wert von `--node-group-count` größer als die aktuelle Anzahl von Shards sein.

Optional können Sie die Availability Zone für jeden Knoten in der Replikationsgruppe mit `--resharding-configuration` angeben.

- `--resharding-configuration` Optional. Eine Liste der bevorzugten Availability Zones für jeden Knoten in jedem Shard in der Replikationsgruppe. Verwenden Sie diesen Parameter nur, wenn der Wert von `--node-group-count` größer ist als die aktuelle Anzahl von Shards. Wenn dieser Parameter beim Hinzufügen von Shards weggelassen wird, ElastiCache wählt Amazon die Availability Zones für die neuen Knoten aus.

Im folgenden Beispiel werden die Keyspaces für vier Shards in einem Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert) mit dem Namen neu konfiguriert. `my-cluster` Das Beispiel gibt auch die Availability Zone für jeden Knoten in jedem Shard an. Die Operation beginnt sofort.

### Example - Shards hinzufügen

Für Linux, macOS oder 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" \
 --apply-immediately
```

```
"PreferredAvailabilityZones=us-east-2b,us-east-2a" \
"PreferredAvailabilityZones=us-east-2c,us-east-2d" \
"PreferredAvailabilityZones=us-east-2d,us-east-2c" \
--apply-immediately
```

Für 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
```

[Weitere Informationen finden Sie unter `-configuration` in der Dokumentation `modify-replication-group-shard`. AWS CLI](#)

Shards hinzufügen (ElastiCache API)

Sie können den verwenden ElastiCache API, um die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) online neu zu konfigurieren, indem Sie den Vorgang verwenden.

`ModifyReplicationGroupShardConfiguration`

Verwenden Sie die folgenden Parameter mit `ModifyReplicationGroupShardConfiguration`.

Parameter

- `ApplyImmediately=true` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `ReplicationGroupId` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `NodeGroupCount` – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Hinzufügen von Shards muss der Wert von `NodeGroupCount` größer als die aktuelle Anzahl von Shards sein.

Optional können Sie die Availability Zone für jeden Knoten in der Replikationsgruppe mit `ReshardingConfiguration` angeben.

- `ReshardingConfiguration` Optional. Eine Liste der bevorzugten Availability Zones für jeden Knoten in jedem Shard in der Replikationsgruppe. Verwenden Sie diesen Parameter nur, wenn der Wert von `NodeGroupCount` größer ist als die aktuelle Anzahl von Shards. Wenn dieser Parameter beim Hinzufügen von Shards weggelassen wird, ElastiCache wählt Amazon die Availability Zones für die neuen Knoten aus.

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mithilfe von `hinzufügen`. ElastiCache API

### Example - Shards hinzufügen

Im folgenden Beispiel werden dem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) Knotengruppen hinzugefügt `my-cluster`, sodass nach Abschluss des Vorgangs insgesamt vier Knotengruppen vorhanden sind. Das Beispiel gibt auch die Availability Zone für jeden Knoten in jedem Shard an. Die Operation beginnt sofort.

```
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>
```

Weitere Informationen finden Sie [ModifyReplicationGroupShardConfiguration](#) in der ElastiCache API Referenz.

## Entfernen von Shards mit Online-Resharding

Sie können Shards mit, oder aus Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) entfernen. AWS Management Console AWS CLI ElastiCache API

### Themen

- [Entfernen von Shards \(Konsole\)](#)
- [Entfernen von Shards \(AWS CLI\)](#)
- [Shards werden entfernt \(\) ElastiCache API](#)

## Entfernen von Shards (Konsole)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mit dem entfernen. AWS Management Console

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, stellen Sie sicher, ElastiCache dass alle Ihre Daten in die verbleibenden Shards passen. Wenn die Daten passen, werden die angegebenen Shards wie angefordert aus der Replikationsgruppe gelöscht. Wenn die Daten nicht in die verbleibenden Knotengruppen passen, wird der Prozess beendet und die Replikationsgruppe hat dieselbe Knotengruppenkonfiguration wie vor dem Zeitpunkt, an dem die Anforderung erstellt wurde.

Sie können den verwenden AWS Management Console , um einen oder mehrere Shards aus Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) zu entfernen. Sie können nicht alle Shards

in einer Replikationsgruppe entfernen. Stattdessen müssen Sie die Replikationsgruppe löschen. Weitere Informationen finden Sie unter [Löschen einer Replikationsgruppe](#). Das folgende Verfahren beschreibt den Vorgang zum Löschen eines oder mehrerer Shards.

Um Shards aus Ihrem Valkey- oder OSS Redis-Cluster (Cluster-Modus aktiviert) zu entfernen

1. Öffnen Sie die Konsole unter ElastiCache . <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.
3. Suchen Sie den Namen des Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert), aus dem Sie Shards entfernen möchten, und wählen Sie ihn aus, nicht das Feld links neben dem Clusternamen.

 Tip

Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) haben in der Spalte Shards einen Wert von 1 oder höher.

4. Wählen Sie in der Liste der Shards das Feld links von den Namen aller Shards, die Sie löschen möchten.
5. Wählen Sie Delete Shard.

### Entfernen von Shards (AWS CLI)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mit dem entfernen. AWS CLI

 Important

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, stellen Sie sicher, ElastiCache dass alle Ihre Daten in die verbleibenden Shards passen. Wenn die Daten passen, werden die angegebenen Shards (`--node-groups-to-remove`) wie angefordert aus der Replikationsgruppe gelöscht und ihre Keyspaces den verbleibenden Shards zugeordnet. Wenn die Daten nicht in die verbleibenden Knotengruppen passen, wird der Prozess beendet und die Replikationsgruppe hat dieselbe Knotengruppenkonfiguration wie vor dem Zeitpunkt, an dem die Anforderung erstellt wurde.

Sie können den verwenden AWS CLI , um einen oder mehrere Shards aus Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) zu entfernen. Sie können nicht alle Shards in einer Replikationsgruppe entfernen. Stattdessen müssen Sie die Replikationsgruppe löschen. Weitere Informationen finden Sie unter [Löschen einer Replikationsgruppe](#).

Verwenden Sie die folgenden Parameter mit `modify-replication-group-shard-configuration`.

#### Parameter

- `--apply-immediately` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.
- `--replication-group-id` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `--node-group-count` – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Entfernen von Shards muss der Wert von `--node-group-count` kleiner als die aktuelle Anzahl von Shards sein.
- `--node-groups-to-remove` Erforderlich, wenn `--node-group-count` kleiner als die aktuelle Anzahl der Knotengruppen (Shards) ist. Eine Liste von Shards (Knotengruppe), die aus der Replikationsgruppe entfernt IDs werden sollen.

Das folgende Verfahren beschreibt den Vorgang zum Löschen eines oder mehrerer Shards.

#### Example - Entfernen von Shards

Im folgenden Beispiel werden zwei Knotengruppen aus dem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) entfernt `my-cluster`, sodass nach Abschluss des Vorgangs insgesamt zwei Knotengruppen vorhanden sind. Die Keyspaces der entfernten Shards werden gleichmäßig auf die verbleibenden Shards verteilt.

Für Linux, macOS oder 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
```

## Für 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
```

## Shards werden entfernt () ElastiCache API

Sie können den verwenden ElastiCache API, um die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) online neu zu konfigurieren, indem Sie den Vorgang verwenden. `ModifyReplicationGroupShardConfiguration`

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mit dem entfernen. ElastiCache API

### Important

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, stellen Sie sicher, ElastiCache dass alle Ihre Daten in die verbleibenden Shards passen. Wenn die Daten passen, werden die angegebenen Shards (`NodeGroupsToRemove`) wie angefordert aus der Replikationsgruppe gelöscht und ihre Keyspaces den verbleibenden Shards zugeordnet. Wenn die Daten nicht in die verbleibenden Knotengruppen passen, wird der Prozess beendet und die Replikationsgruppe hat dieselbe Knotengruppenkonfiguration wie vor dem Zeitpunkt, an dem die Anforderung erstellt wurde.

Sie können den verwenden ElastiCache API, um einen oder mehrere Shards aus Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) zu entfernen. Sie können nicht alle Shards in einer Replikationsgruppe entfernen. Stattdessen müssen Sie die Replikationsgruppe löschen. Weitere Informationen finden Sie unter [Löschen einer Replikationsgruppe](#).

Verwenden Sie die folgenden Parameter mit `ModifyReplicationGroupShardConfiguration`.

### Parameter

- `ApplyImmediately=true` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.

- `ReplicationGroupId` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `NodeGroupCount` – Erforderlich. Gibt die Anzahl der Shards (Knotengruppen) an, die nach Abschluss des Vorgangs vorhanden sein müssen. Beim Entfernen von Shards muss der Wert von `NodeGroupCount` kleiner als die aktuelle Anzahl von Shards sein.
- `NodeGroupsToRemove` Erforderlich, wenn `--node-group-count` kleiner als die aktuelle Anzahl der Knotengruppen (Shards) ist. Eine Liste von Shards (Knotengruppe), die aus der Replikationsgruppe entfernt IDs werden sollen.

Das folgende Verfahren beschreibt den Vorgang zum Löschen eines oder mehrerer Shards.

### Example - Entfernen von Shards

Im folgenden Beispiel werden zwei Knotengruppen aus dem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) entfernt `my-cluster`, sodass nach Abschluss des Vorgangs insgesamt zwei Knotengruppen vorhanden sind. Die Keyspaces der entfernten Shards werden gleichmäßig auf die verbleibenden Shards verteilt.

```
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>
```

### Online-Shard-Rebalancing

Sie können Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) mithilfe von, oder neu verteilen. AWS Management Console AWS CLI ElastiCache API

### Themen

- [Online-Shard-Rebalancing \(Console\)](#)
- [Online-Shard-Rebalancing \(AWS CLI\)](#)

- [Online-Shard-Rebalancing \(ElastiCache API\)](#)

## Online-Shard-Rebalancing (Console)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie die Shards mithilfe von neu verteilen. AWS Management Console

Um die Keyspaces zwischen den Shards auf Ihrem Valkey- oder Redis-Cluster (Clustermodus aktiviert) neu auszurichten OSS

1. Öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.
3. Wählen Sie den Namen des Valkey- oder Redis-Clusters OSS (Clustermodus aktiviert), den Sie neu verteilen möchten, und nicht das Feld links neben dem Namen.

 Tip

Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) haben in der Spalte Shards einen Wert von 1 oder höher.

4. Wählen Sie Rebalance aus.
5. Wählen Sie nach Aufforderung die Option Rebalance aus. Möglicherweise wird eine Meldung ähnlich der folgenden angezeigt: *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)*. Wenn Sie dies tun, wählen Sie Abbrechen.

## Online-Shard-Rebalancing (AWS CLI)

Verwenden Sie die folgenden Parameter mit `modify-replication-group-shard-configuration`.

### Parameter

- `-apply-immediately` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.

- `--replication-group-id` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `--node-group-count` – Erforderlich. Um die Keyspaces über alle Shards im Cluster neu zu verteilen, muss dieser Wert der aktuellen Anzahl von Shards entsprechen.

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie die Shards mithilfe von neu verteilen. AWS CLI

### Example - Ausgleichen der Shards in einem Cluster

Im folgenden Beispiel werden die Steckplätze im Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu verteilt, sodass die Steckplätze `my-cluster` so gleichmäßig wie möglich verteilt werden. Der Wert von `--node-group-count` (4) ist die Anzahl der derzeit im Cluster befindlichen Shards.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group-shard-configuration \
 --replication-group-id my-cluster \
 --node-group-count 4 \
 --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group-shard-configuration ^
 --replication-group-id my-cluster ^
 --node-group-count 4 ^
 --apply-immediately
```

### Online-Shard-Rebalancing (ElastiCache API)

Sie können den verwenden ElastiCache API, um die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) online neu zu konfigurieren, indem Sie den Vorgang verwenden.

#### ModifyReplicationGroupShardConfiguration

Verwenden Sie die folgenden Parameter mit `ModifyReplicationGroupShardConfiguration`.

#### Parameter

- `ApplyImmediately=true` – Erforderlich. Gibt an, dass die Shard-Rekonfigurationsoperation sofort gestartet werden soll.

- `ReplicationGroupId` – Erforderlich. Gibt an, für welche Replikationsgruppe (Cluster) die Shard-Rekonfigurationsoperation ausgeführt werden soll.
- `NodeGroupCount` – Erforderlich. Um die Keyspaces über alle Shards im Cluster neu zu verteilen, muss dieser Wert der aktuellen Anzahl von Shards entsprechen.

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu konfigurieren, indem Sie die Shards mithilfe von neu verteilen.

ElastiCache API

### Example - Cluster-Ausgleich

Im folgenden Beispiel werden die Steckplätze im Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert) neu verteilt, sodass die Steckplätze `my-cluster` so gleichmäßig wie möglich verteilt werden. Der Wert von `NodeGroupCount` (4) ist die Anzahl der derzeit im Cluster befindlichen Shards.

```
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>
```

### Vertikales Online-Skalieren durch Ändern des Knotentyps

Durch die Verwendung der vertikalen Online-Skalierung mit Valkey Version 7.2 oder neuer oder Redis OSS Version 3.2.10 oder neuer können Sie Ihre Valkey- oder OSS Redis-Cluster dynamisch mit minimalen Ausfallzeiten skalieren. Dadurch kann Ihr Valkey- oder OSS Redis-Cluster Anfragen auch während der Skalierung bearbeiten.

#### Note

Die Skalierung zwischen einem Cluster mit Daten-Tiering (z. B. ein Cluster, der einen R6gd-Knotentyp verwendet) und einem Cluster ohne Daten-Tiering (z. B. ein Cluster, der einen

R6g-Knotentyp verwendet) wird nicht unterstützt. Weitere Informationen finden Sie unter [Daten-Tiering ElastiCache](#).

Sie haben die folgenden Möglichkeiten:

- Skalieren — Erhöhen Sie die Lese- und Schreibkapazität, indem Sie den Knotentyp Ihres Valkey- oder OSS Redis-Clusters so anpassen, dass ein größerer Knotentyp verwendet wird.

ElastiCache passt die Größe Ihres Clusters dynamisch an, bleibt aber online und bearbeitet Anfragen.

- Herunterskalierung – Verringern Sie die Lese- und Schreibkapazität, indem Sie den Knotentyp auf die Verwendung eines kleineren Knotens anpassen. Auch hier wird die Größe Ihres Clusters ElastiCache dynamisch angepasst, während er online bleibt und Anfragen bearbeitet. In diesem Fall verringern Sie die Kosten durch die Verkleinerung des Knotens.

#### Note

Aufwärts- und Abwärtsskalieren basiert auf der Erstellung von neu ausgewählten Knotentypen und der Synchronisierung der neuen Knoten mit den vorherigen. Um reibungsloses Abwärts-/Aufwärtsskalieren zu gewährleisten, führen Sie Folgendes durch:

- Stellen Sie sicher, dass Sie über ausreichende Kapazität ENI (Elastic Network Interface) verfügen. Vergewissern Sie sich beim Abwärtsskalieren, dass der kleinere Knoten über ausreichend Speicher für erwarteten Datenverkehr verfügt.

Erfahren Sie mehr über bewährte Methoden für die Speicherverwaltung unter [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#).

- Während die vertikale Skalierung ausgelegt ist, vollständig online zu bleiben, basiert sie auf der Synchronisierung von Daten zwischen dem alten und dem neuen Knoten. Wir empfehlen, dass Sie Abwärts-/Aufwärtsskalieren zu einem Zeitpunkt durchführen, an dem der Datenverkehr am geringsten ist.
- Testen Sie das Verhalten Ihrer Anwendung während der Skalierung möglichst in einer Staging-Umgebung.

## Inhalt

- [Online-aufwärtsskalieren](#)
  - [Skalierung von Valkey- oder OSS Redis-Cache-Clustern \(Konsole\)](#)
  - [Skalieren von Valkey- oder OSS Redis-Cache-Clustern \(\)AWS CLI](#)
  - [Skalierung von Valkey- oder OSS Redis-Cache-Clustern \(\) ElastiCache API](#)
- [Online-abwärtsskalieren](#)
  - [Herunterskalierung der Valkey- oder OSS Redis-Cache-Cluster \(Konsole\)](#)
  - [Herunterskalieren von Valkey- oder OSS Redis-Cache-Clustern \(\)AWS CLI](#)
  - [Herunterskalieren der Valkey- oder OSS Redis-Cache-Cluster \(\) ElastiCache API](#)

## Online-aufwärtsskalieren

### Themen

- [Skalierung von Valkey- oder OSS Redis-Cache-Clustern \(Konsole\)](#)
- [Skalieren von Valkey- oder OSS Redis-Cache-Clustern \(\)AWS CLI](#)
- [Skalierung von Valkey- oder OSS Redis-Cache-Clustern \(\) ElastiCache API](#)

## Skalierung von Valkey- oder OSS Redis-Cache-Clustern (Konsole)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder OSS Redis-Cluster mithilfe der ElastiCache Management Console skalieren. Während dieses Vorgangs bearbeitet Ihr Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Um einen Valkey- oder OSS Redis-Cluster (Konsole) zu skalieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.
3. Wählen Sie aus der Liste der Cluster den Cluster aus.
4. Wählen Sie Ändern aus.
5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
  - Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten. Wählen Sie zur Aufwärtsskalierung einen Knotentyp, der größer als Ihr bestehender Knoten ist.

6. Wenn Sie die Aufwärtsskalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen `Apply immediately`. Wenn Sie das Kontrollkästchen für `Apply immediately` nicht aktivieren, wird die Skalierung während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie `Ändern` aus.

Wenn Sie im vorherigen Schritt `Apply immediately` aktiviert haben, ändert sich der Cluster-Status zu `modifying`. Wenn der Status zu `available` wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

## Skalieren von Valkey- oder OSS Redis-Cache-Clustern (AWS CLI)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder OSS Redis-Cache-Cluster mithilfe von skalieren. AWS CLI Während dieses Vorgangs bearbeitet Ihr Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

### Um einen Valkey- oder OSS Redis-Cache-Cluster zu skalieren (AWS CLI)

1. Ermitteln Sie die Knotentypen, auf die Sie skalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-replication-group-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
 --replication-group-id my-replication-group-id
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

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

Weitere Informationen finden Sie unter [list-allowed-node-type-Änderungen](#) in der AWS CLI Referenz.

2. Ändern Sie Ihre Replikationsgruppe mithilfe des AWS CLI `modify-replication-group` Befehls und der folgenden Parameter, sodass sie auf den neuen, größeren Knotentyp skaliert werden kann.
  - `--replication-group-id` Der Name der zu erstellenden Replikationsgruppe, zu der Sie aufwärtsskalieren.
  - `--cache-node-type` Der neue Knotentyp, auf den der Cache-Cluster skaliert werden soll. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
  - `--cache-parameter-group-name` – [Optional) Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
  - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id my-redis-cluster \
 --cache-node-type cache.m3.xlarge \
 --apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id my-redis-cluster ^
 --cache-node-type cache.m3.xlarge ^
 --apply-immediately
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

```
{
 "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.m3.xlarge",
 "DataTiering": "disabled"
 "PendingModifiedValues": {}
}
}
```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des Cache-Clusters mithilfe des AWS CLI `describe-cache-clusters` Befehls mit dem folgenden Parameter. Wenn der Status zu `available` wechselt, können Sie den neuen, größeren Cache-Cluster-Knoten verwenden.

## Skalierung von Valkey- oder OSS Redis-Cache-Clustern () ElastiCache API

Der folgende Prozess skaliert Ihren Cache-Cluster von seinem aktuellen Knotentyp auf einen neuen, größeren Knotentyp mithilfe von ElastiCache API. Während dieses Vorgangs werden die DNS Einträge ElastiCache aktualisiert, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Für Valkey 7.2 und höher, Redis OSS 5.0.5 und höher, können Sie Cluster mit auto Failover skalieren, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Bei Version Redis OSS 4.0.10 und niedriger stellen Sie möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge auf früheren Versionen vom Primärknoten aus fest, während der Eintrag aktualisiert wird. DNS

Wie lange die Aufwärtsskalierung auf einen größeren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

## So skalieren Sie einen Valkey- oder OSS Redis-Cache-Cluster () ElastiCache API

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache API `ListAllowedNodeTypeModifications` Aktion mit dem folgenden Parameter skalieren können.

- `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [ListAllowedNodeTypeModifications](#) in der ElastiCache API Amazon-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyReplicationGroup` ElastiCache API Aktion und mit den folgenden Parametern auf den neuen Knotentyp.
  - `ReplicationGroupId` der Name der Replikationsgruppe.
  - `CacheNodeType` der neue, größere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Dieser Wert muss einer der Instanztypen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
  - `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
  - `ApplyImmediately` Legen Sie `true` fest, um die Skalierung sofort auszuführen. Wenn Sie die Skalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `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>
```

Weitere Informationen finden Sie unter [ModifyReplicationGroup](#) in der ElastiCache API Amazon-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überwachen Sie den Status der Replikationsgruppe mithilfe der ElastiCache API `DescribeReplicationGroups` Aktion mit den folgenden Parametern. Wenn der Status von `modifying` zu `available` wechselt, können Sie die neue, aufskalierte Replikationsgruppe verwenden.
  - `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

```
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>
```

Weitere Informationen finden Sie unter [DescribeReplicationGroups](#) in der ElastiCache API Amazon-Referenz.

## Online-abwärtsskalieren

### Themen

- [Herunterskalierung der Valkey- oder OSS Redis-Cache-Cluster \(Konsole\)](#)
- [Herunterskalieren von Valkey- oder OSS Redis-Cache-Clustern \(AWS CLI\)](#)
- [Herunterskalieren der Valkey- oder OSS Redis-Cache-Cluster \(ElastiCache API\)](#)

### Herunterskalierung der Valkey- oder OSS Redis-Cache-Cluster (Konsole)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder OSS Redis-Cluster mithilfe der ElastiCache Management Console herunterskalieren. Während dieses Vorgangs bearbeitet Ihr Valkey- oder OSS Redis-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

### Um einen Valkey- oder OSS Redis-Cluster herunterzuskalieren (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.
3. Wählen Sie aus der Liste der Cluster Ihren bevorzugten Cluster.
4. Wählen Sie Ändern aus.
5. Gehen Sie im Modify Cluster-Assistenten wie folgt vor:
  - Wählen Sie in der Liste Node type den Knotentyp aus, auf den Sie skalieren möchten. Wählen Sie zur Abwärtsskalierung einen Knotentyp, der kleiner als Ihr bestehender Knoten ist. Beachten Sie, dass nicht alle Knotentypen für das Herunterskalieren zur Verfügung stehen.
6. Wenn Sie die Abwärtsskalierung sofort starten möchten, aktivieren Sie das Kontrollkästchen Apply immediately. Wenn Sie das Kontrollkästchen für Apply immediately nicht aktivieren, wird die Abwärtsskalierung während des nächsten Wartungsfensters des Clusters durchgeführt.
7. Wählen Sie Ändern aus.

Wenn Sie im vorherigen Schritt Apply immediately aktiviert haben, ändert sich der Cluster-Status zu modifying. Wenn der Status zu available wechselt, ist die Änderung abgeschlossen und Sie können den neuen Cluster verwenden.

## Herunterskalieren von Valkey- oder OSS Redis-Cache-Clustern (AWS CLI)

Das folgende Verfahren beschreibt, wie Sie einen Valkey- oder OSS Redis-Cache-Cluster mithilfe von herunterskalieren. AWS CLI Während dieses Vorgangs bearbeitet Ihr Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

### Um einen Valkey- oder OSS Redis-Cache-Cluster herunterzuskalieren (AWS CLI)

1. Ermitteln Sie die Knotentypen, auf die Sie herunterskalieren können, indem Sie den AWS CLI `list-allowed-node-type-modifications` Befehl mit dem folgenden Parameter ausführen.

Für Linux, macOS oder Unix:

```
aws elasticache list-allowed-node-type-modifications \
 --replication-group-id my-replication-group-id
```

Für Windows:

```
aws elasticache list-allowed-node-type-modifications ^
 --replication-group-id my-replication-group-id
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

```
{
 "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"
]
}
```

Weitere Informationen finden Sie unter [list-allowed-node-type-Änderungen](#) in der AWS CLI Referenz.

2. Ändern Sie Ihre Replikationsgruppe mithilfe des AWS CLI `modify-replication-group` Befehls und der folgenden Parameter, sodass sie auf den neuen, kleineren Knotentyp herunterskaliert wird.
  - `--replication-group-id` Der Name der Replikationsgruppe, zu der Sie abwärts skalieren.
  - `--cache-node-type` Der neue Knotentyp, auf den der Cache-Cluster skaliert werden soll. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Befehl `list-allowed-node-type-modifications` zurückgegeben wurden.
  - `--cache-parameter-group-name` – [Optional) Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
  - `--apply-immediately` – Führt die Skalierung sofort aus. Wenn Sie die Skalierung für das nächste Wartungsfenster des Clusters planen möchten, verwenden Sie den Parameter `--no-apply-immediately`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id my-redis-cluster \
 --cache-node-type cache.t2.micro \
```

```
--apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id my-redis-cluster ^
 --cache-node-type cache.t2.micro ^
 --apply-immediately
```

Die Ausgabe des obigen Befehls sieht ungefähr so aus (JSONFormat).

```
{
 "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": {}
 }
}
```

Weitere Informationen finden Sie unter [modify-replication-group](#) in der AWS CLI Referenz.

3. Wenn Sie den verwendet haben `--apply-immediately`, überprüfen Sie den Status des Cache-Clusters mithilfe des AWS CLI `describe-cache-clusters` Befehls mit dem folgenden Parameter. Wenn der Status zu `available` wechselt, können Sie den neuen, kleineren Cache-Cluster-Knoten verwenden.

## Herunterskalieren der Valkey- oder OSS Redis-Cache-Cluster () ElastiCache API

Der folgende Prozess skaliert Ihre Replikationsgruppe von ihrem aktuellen Knotentyp auf einen neuen, kleineren Knotentyp mithilfe von. ElastiCache API Während dieses Vorgangs bearbeitet Ihr Valkey- oder OSS Redis-Cluster weiterhin Anfragen mit minimaler Ausfallzeit.

Wie lange die Abwärtsskalierung auf einen kleineren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

## Herunterskalieren () ElastiCache API

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache API `ListAllowedNodeTypeModifications` Aktion mit dem folgenden Parameter herunterskalieren können.
  - `ReplicationGroupId` der Name der Replikationsgruppe. Verwenden Sie diesen Parameter, um eine bestimmte Replikationsgruppe zu beschreiben, statt alle Replikationsgruppen anzugeben.

<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>
```

Weitere Informationen finden Sie unter [ListAllowedNodeTypeModifications](#) in der ElastiCache API Amazon-Referenz.

2. Skalieren Sie Ihre aktuelle Replikationsgruppe mithilfe der `ModifyReplicationGroup` ElastiCache API Aktion und mit den folgenden Parametern auf den neuen Knotentyp herunter.
  - `ReplicationGroupId` der Name der Replikationsgruppe.
  - `CacheNodeType` Der neue, kleinere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Dieser Wert muss einer der Instanztypen sein, die von der `ListAllowedNodeTypeModifications` Aktion im vorherigen Schritt zurückgegeben wurden.
  - `CacheParameterGroupName` – [Optional] Verwenden Sie diesen Parameter, wenn Sie `reserved-memory` verwenden, um Speicher für den Cluster zu reservieren. Geben Sie eine benutzerdefinierte Cache-Parametergruppe an, die die korrekte Speichermenge für den neuen Knotentyp reserviert. Wenn Sie `reserved-memory-percent` verwenden, können Sie diesen Parameter weglassen.
  - `ApplyImmediately` – Legen Sie `true` fest, um die Abwärtsskalierung sofort auszuführen. Wenn Sie die Abwärtsskalierung für das nächste Wartungsfenster planen möchten, verwenden Sie `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>
```

Weitere Informationen finden Sie unter [ModifyReplicationGroup](#) in der ElastiCache API Amazon-Referenz.

## Erste Schritte mit JSON für Valkey und Redis OSS

ElastiCache unterstützt das native Format JavaScript Object Notation (JSON), eine einfache, schemalose Methode zur Kodierung komplexer Datensätze in Valkey- und Redis-Clustern. OSS Sie können Daten mithilfe des JavaScript Object Notation (JSON) -Formats nativ innerhalb der Cluster speichern und darauf zugreifen und die in diesen Clustern gespeicherten JSON Daten aktualisieren, ohne dass Sie benutzerdefinierten Code für die Serialisierung und Deserialisierung verwalten müssen.

Sie können jetzt nicht nur Valkey- und OSS API Redis-Operationen für Anwendungen verwenden, die darüber arbeiten JSON, sondern auch bestimmte Teile eines Dokuments effizient abrufen und aktualisieren, ohne das gesamte Objekt bearbeiten zu müssen. JSON Dies kann die Leistung verbessern und Kosten senken. [Sie können den Inhalt Ihres JSON Dokuments auch mithilfe der Abfrage im Goessner-Stil durchsuchen.](#) JSONPath

Nachdem Sie einen Cluster mit einer unterstützten Engine-Version erstellt haben, sind der JSON Datentyp und die zugehörigen Befehle automatisch verfügbar. API kompatibel und RDB kompatibel mit Version 2 des JSON Moduls, sodass Sie bestehende Valkey- und JSON OSS Redis-Anwendungen problemlos in diese Version migrieren können. ElastiCache Weitere Informationen zu den unterstützten Befehlen finden Sie unter [Unterstützte Valkey- und Redis-Befehle OSS](#)

Die JSON zugehörigen Messwerte `JsonBasedCmds` und `JsonBasedCmdsLatency` dienen CloudWatch zur Überwachung der Verwendung dieses Datentyps. Weitere Informationen finden Sie unter [Metriken für Valkey und OSS Redis](#).

### Note

Zur Verwendung JSON müssen Sie Valkey 7.2 und höher oder Redis OSS Engine-Version 6.2.6 oder höher ausführen.

## Themen

- [JSONÜberblick über Datentypen](#)
- [Unterstützte Valkey- und Redis-Befehle OSS](#)

## JSONÜberblick über Datentypen

ElastiCache unterstützt eine Reihe von Valkey- und OSS Redis-Befehlen für die Arbeit mit dem JSON Datentyp. Im Folgenden finden Sie eine Übersicht über den JSON Datentyp und eine detaillierte Liste der unterstützten Befehle.

## Terminologie

Begriff	Beschreibung
JSONDokument	Bezieht sich auf den Wert eines JSON Schlüssels.
JSONWert	Bezieht sich auf eine Teilmenge eines JSON Dokuments, einschließlich der Wurzel, die das gesamte Dokument darstellt. Ein Wert könnte ein Container oder ein Eintrag innerhalb eines Containers sein.
JSONElement	Entspricht dem JSON Wert.

## Unterstützter JSON Standard

JSONDas Format entspricht den [RFCJSONDatenaustauschstandards 7159](#) und [ECMA-404](#). UTF-8 [Unicode](#) im JSON Text wird unterstützt.

## Stammelement

Das Stammelement kann einen beliebigen JSON Datentyp haben. Beachten Sie, dass in früheren Versionen der Version RFC 4627 nur Objekte oder Arrays als Stammwerte zulässig waren. Seit der Aktualisierung auf Version RFC 7159 kann das Stammverzeichnis eines JSON Dokuments einen beliebigen JSON Datentyp haben.

## Begrenzung der Dokumentgröße

JSONDokumente werden intern in einem Format gespeichert, das für schnellen Zugriff und Änderungen optimiert ist. Dieses Format führt in der Regel dazu, dass etwas mehr Speicher verbraucht wird als bei der äquivalenten serialisierten Darstellung desselben Dokuments.

Der Speicherverbrauch eines einzelnen JSON Dokuments ist auf 64 MB begrenzt. Dies entspricht der Größe der speicherinternen Datenstruktur, nicht der JSON Zeichenfolge. Mit dem `JSON.DEBUG MEMORY` Befehl können Sie überprüfen, wie viel Speicherplatz ein JSON Dokument belegt.

## JSON ACLs

- Ähnlich wie bei den bestehenden Kategorien pro Datentyp (`@string`, `@hash` usw.) wurde eine neue Kategorie `@json` hinzugefügt, um die Verwaltung des Zugriffs auf JSON Befehle und Daten zu vereinfachen. Keine anderen vorhandenen Valkey- oder OSS Redis-Befehle gehören zur Kategorie `@json`. Alle JSON Befehle setzen alle Einschränkungen und Berechtigungen für Keyspace oder Befehle durch.
- Es gibt fünf bestehende Valkey- und OSS ACL Redis-Kategorien, die aktualisiert wurden und nun die neuen JSON Befehle enthalten: `@read`, `@write`, `@fast`, `@slow` und `@admin`. Die folgende Tabelle zeigt die Zuordnung von JSON Befehlen zu den entsprechenden Kategorien.

## ACL

JSONBefehl	@read	@write	@fast	@slow	@admin
JSON.ARRAPPEND		y	y		
JSON.ARRINDEX	y		y		
JSON.ARRINSERT		y	y		
JSON.ARRLEN	y		y		
JSON.ARRPOP		y	y		

JSONBefehl	@read	@write	@fast	@slow	@admin
JSON.ARRTRIM		y	y		
JSON.CLEAR		y	y		
JSON.DEBUG	y			y	y
JSON.DEL		y	y		
JSON.FORGET		y	y		
JSON.GET	y		y		
JSON.MGET	y		y		
JSON.NUMINCRBY		y	y		
JSON.NUMMULTBY		y	y		
JSON.OBJECTS	y		y		
JSON.OBJECTLEN	y		y		
JSON.RESP	y		y		
JSON.SET		y		y	
JSON.STRAPPEND		y	y		

JSONBefehl	@read	@write	@fast	@slow	@admin
JSON.STRL EN	y		y		
JSON.STRL EN	y		y		
JSON.TOGG LE		y	y		
JSON.TYPE	y		y		
JSON.NUMI NCRBY		y	y		

## Begrenzung der Verschachtelungstiefe

Wenn ein JSON Objekt oder eine Reihe ein Element hat, das selbst ein anderes JSON Objekt oder Array ist, wird gesagt, dass dieses innere Objekt oder Array innerhalb des äußeren Objekts oder der äußeren Anordnung „verschachtelt“ ist. Die maximale Verschachtelungstiefe ist 128. Jeder Versuch, ein Dokument zu erstellen, das eine Verschachtelungstiefe von mehr als 128 enthält, wird mit einem Fehler abgelehnt.

## Befehlssyntax

Die meisten Befehle erfordern einen Schlüsselnamen als erstes Argument. Einige Befehle haben auch ein Pfadargument. Das Pfadargument ist standardmäßig das Stammverzeichnis, wenn es optional und nicht im Lieferumfang enthalten ist.

Notation:

- Erforderliche Argumente sind in spitzen Klammern eingeschlossen. Zum Beispiel: <key>
- Optionale Argumente werden in eckige Klammern gesetzt. Zum Beispiel: [path]
- Zusätzliche optionale Argumente sind durch eine Ellipse („...“) gekennzeichnet. Zum Beispiel: [json ...]

## Pfadsyntax

Redis JSON unterstützt zwei Arten von Pfadsyntaxen:

- Verbesserte Syntax — Folgt der von [Goessner](#) beschriebenen JSONPath Syntax, wie in der folgenden Tabelle dargestellt. Wir haben die Beschreibungen in der Tabelle zur besseren Übersicht neu angeordnet und geändert.
- Beschränkte Syntax – Hat begrenzte Abfragemöglichkeiten.

### Note

Die Ergebnisse einiger Befehle sind davon abhängig, welche Art von Pfadsyntax verwendet wird.

Wenn ein Abfragepfad mit „\$“ beginnt, verwendet er die erweiterte Syntax. Andernfalls wird eine eingeschränkte Syntax verwendet.

### Verbesserte Syntax

Symbol/Ausdruck	Beschreibung
\$	Das Stammelement.
. oder []	Untergeordneter Operator
..	Rekursiver Abstieg
*	Platzhalter Alle Elemente in einem Objekt oder Array.
[]	Array-Index-Operator Der Index basiert auf 0.
[,]	Union-Operator
[start:end:step]	Array-Slice-Operator
?()	Wendet einen Filterausdruck (Skript) auf das aktuelle Array oder Objekt an.

Symbol/Ausdruck	Beschreibung
()	Filterausdruck
@	Wird in Filterausdrücken verwendet, die zum aktuellen Knoten verweisen, der verarbeitet wird.
==	Ist gleich; wird in Filterausdrücken verwendet.
!=	Ist nicht gleich; wird in Filterausdrücken verwendet.
>	Größer als; wird in Filterausdrücken verwendet.
>=	Größer als oder gleich; wird in Filterausdrücken verwendet.
<	Kleiner als; wird in Filterausdrücken verwendet.
<=	Kleiner als oder gleich; wird in Filterausdrücken verwendet.
&&	LogischAND, wird verwendet, um mehrere Filterausdrücke zu kombinieren.
	Logisches ODER; wird verwendet, um mehrere Filterausdrücke zu kombinieren.

## Beispiele

Die folgenden Beispiele basieren auf den XML Beispieldaten [von Goessner](#), die wir durch das Hinzufügen zusätzlicher Felder modifiziert haben.

```
{ "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
}
}
```

Pfad	Beschreibung
<code>\$.store.book[*].author</code>	Die Autoren aller Bücher im Laden.
<code>\$.author</code>	Alle Autoren

Pfad	Beschreibung
<code>\$.store.*</code>	Alle Mitglieder des Ladens.
<code>\$["store"].*</code>	Alle Mitglieder des Ladens.
<code>\$.store..price</code>	Der Preis von allem im Laden.
<code>\$..*</code>	Alle rekursiven Mitglieder der Struktur. JSON
<code>\$.book[*]</code>	Alle Bücher.
<code>\$.book[0]</code>	Das erste Buch.
<code>\$.book[-1]</code>	Das letzte Buch.
<code>\$.book[0:2]</code>	Die ersten beiden Bücher.
<code>\$.book[0,1]</code>	Die ersten beiden Bücher.
<code>\$.book[0:4]</code>	Bücher von Index 0 bis 3 (Endindex ist nicht inklusive).
<code>\$.book[0:4:2]</code>	Bücher bei Index 0, 2.
<code>\$.book[?(@.isbn)]</code>	Alle Bücher mit einer ISBN Nummer.
<code>\$.book[?(@.price&lt;10)]</code>	Alle Bücher günstiger als 10 USD.
<code>'\$.book[?(@.price &lt; 10)]'</code>	Alle Bücher günstiger als 10 USD. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.)
<code>'\$.book[?(@"price" &lt; 10)]'</code>	Alle Bücher günstiger als 10 USD.
<code>'\$.book[?(@"price" &lt; 10)]'</code>	Alle Bücher günstiger als 10 USD.
<code>\$.book[?(@.price&gt;=10&amp;&amp;@.price&lt;=100)]</code>	Alle Bücher im Preisbereich von 10 bis 100 USD inklusive.

Pfad	Beschreibung
'\$.book[?(@.price>=10 && @.price<=100)]'	Alle Bücher im Preisbereich von 10 bis 100 USD inklusive. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.)
\$.book[?(@.sold==true  @.in-stock==false)]	Alle Bücher verkauft oder ausverkauft.
'\$.book[?(@.sold == true    @.in-stock == false)]'	Alle Bücher verkauft oder ausverkauft. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.)
'\$.store.book[?(@.["category"] == "fiction")]'	Alle Bücher der Kategorie Belletristik.
'\$.store.book[?(@.["category"] != "fiction")]'	Alle Bücher in der Kategorie Sachbücher.

### Zusätzliche Beispiele für Filterausdrücke:

```

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]"
```

## Beschränkte Syntax

Symbol/Ausdruck	Beschreibung
. oder []	Untergeordneter Operator
[]	Array-Index-Operator Der Index basiert auf 0.

## Beispiele

Pfad	Beschreibung
.store.book[0].author	Der Autor des ersten Buches.
.store.book[-1].author	Der Autor des letzten Buches.
.address.city	Name der Stadt.
["store"]["book"][0]["title"]	Der Titel des ersten Buches.
["store"]["book"][-1]["title"]	Der Titel des letzten Buches.

### Note

Alle [Goessner](#)-Inhalte, die in dieser Dokumentation erwähnt werden, unterliegen der [Creative-Commons-Lizenz](#).

## Häufige Fehlerpräfixe

Jede Fehlermeldung hat ein Präfix. Im Folgenden finden Sie eine Liste mit allgemeinen Fehlerpräfixen.

Präfix	Beschreibung
ERR	Ein allgemeiner Fehler.
LIMIT	Ein Fehler, der auftritt, wenn die Größenbeschränkung überschritten wird. Zum Beispiel wurde die Größenbeschränkung oder Verschachtelungstiefe überschritten.
NONEXISTENT	Ein Schlüssel oder Pfad ist nicht vorhanden.
OUTOFBOUNDARIES	Array-Index außerhalb des gültigen Bereichs.
SYNTAXERR	Syntaxfehler
WRONGTYPE	Falscher Werttyp.

## JSONverwandte Metriken

Die folgenden JSON Informationsmetriken werden bereitgestellt:

Informationen	Beschreibung
json_total_memory_bytes	Gesamter Speicher, der JSON Objekten zugewiesen ist.
json_num_documents	Gesamtzahl der Dokumente in Valkey oder OSS Redis.

Führen Sie den folgenden Befehl aus, um Kernmetriken abzufragen:

```
info json_core_metrics
```

## Wie ElastiCache interagiert Valkey und Redis OSS mit JSON

Im folgenden Abschnitt wird beschrieben, wie Valkey und Redis ElastiCache mit dem Datentyp OSS interagieren. JSON

### Rangfolge der Operatoren

Bei der Bewertung bedingter Ausdrücke zum Filtern, haben `&&`s zuerst Vorrang, und dann werden `||`s ausgewertet, wie es in den meisten Sprachen üblich ist. Operationen innerhalb von Klammern werden zuerst ausgeführt.

### Verhalten der maximalen Verschachtelungsbeschränkung

Das maximale Limit für die Verschachtelung von Pfaden in ElastiCache (OSSRedis) beträgt 128. Ein Wert wie `$.a.b.c.d...` kann also nur 128 Level erreichen.

### Umgang mit numerischen Werten

JSON hat keine separaten Datentypen für Ganzzahlen und Fließkommazahlen. Sie werden alle Zahlen genannt.

### Numerische Repräsentationen:

Wenn eine JSON Zahl bei der Eingabe empfangen wird, wird sie in eine der beiden internen Binärdarstellungen umgewandelt: eine 64-Bit-Ganzzahl mit Vorzeichen oder eine 64-Bit-Gleitkommazahl IEEE mit doppelter Genauigkeit. Die Ursprüngliche Zeichenfolge und alle ihre Formatierungen werden nicht beibehalten. Wenn also eine Zahl als Teil einer JSON Antwort ausgegeben wird, wird sie aus der internen Binärdarstellung in eine druckbare Zeichenfolge konvertiert, die generische Formatierungsregeln verwendet. Diese Regeln könnten dazu führen, dass eine andere Zeichenfolge generiert wird als empfangen wurde.

### Arithmetische Befehle NUMINCRBY und NUMMULTBY:

- Wenn beide Zahlen ganze Zahlen sind und das Ergebnis außerhalb des Bereichs von `int64` liegt, wird es automatisch zu einer 64-Bit-Gleitkommazahl IEEE mit doppelter Genauigkeit.
- Wenn mindestens eine der Zahlen eine Fließkommazahl ist, ist das Ergebnis eine 64-Bit-Gleitkommazahl IEEE mit doppelter Genauigkeit.
- Wenn das Ergebnis den Bereich von 64 Bit (IEEEdoppelt) überschreitet, gibt der Befehl einen `OVERFLOW` Fehler zurück.

Eine detaillierte Liste der verfügbaren Befehle finden Sie unter [Unterstützte Valkey- und Redis-Befehle OSS](#).

## Direktes Array-Filtern

ElastiCache mit Valkey oder Redis werden Array-Objekte direkt OSS gefiltert.

Bei Daten wie `[0, 1, 2, 3, 4, 5, 6]` und einer Pfadabfrage wie `$[?(@<4)]` oder Daten wie `{"my_key": [0, 1, 2, 3, 4, 5, 6]}` und einer Pfadabfrage wie `$.my_key[?(@<4)]` ElastiCache mit Valkey oder Redis OSS würde unter beiden Umständen `[1,2,3]` zurückgegeben werden.

## Array-Indizierung

ElastiCache mit Valkey oder Redis sind sowohl positive als auch OSS negative Indizes für Arrays möglich. Bei einem Array mit der Länge fünf würde 0 das erste Element abfragen, 1 das zweite usw. Negative Zahlen beginnen am Ende des Arrays, also würde -1 das fünfte Element abfragen, -2 das vierte Element usw.

Um ein vorhersehbares Verhalten für Kunden sicherzustellen, ElastiCache rundet Valkey oder Redis Array-Indizes OSS nicht nach unten oder oben ab. Wenn Sie also ein Array mit einer Länge von 5 haben, würde der Aufruf von Index 5 oder höher oder -6 oder niedriger zu keinem Ergebnis führen.

## Strikte Syntaxbewertung

MemoryDB erlaubt keine JSON Pfade mit ungültiger Syntax, selbst wenn eine Teilmenge des Pfads einen gültigen Pfad enthält. Dies soll für unsere Kunden ein korrektes Verhalten sicherstellen.

# Unterstützte Valkey- und Redis-Befehle OSS

ElastiCache unterstützt die folgenden Valkey- und Redis-Befehle OSSJSON:

## Themen

- [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

Hängen Sie einen oder mehrere Werte für die Array-Werte an den Pfad an.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- **path** (erforderlich) — Ein JSON Pfad.
- **json** (erforderlich) — Der JSON Wert, der an das Array angehängt werden soll.

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.
- SYNTAXERRFehler, wenn eines der JSON-Eingabeargumente keine gültige JSON Zeichenfolge ist.
- NONEXISTENT-Fehler, wenn der Pfad nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge des Arrays.
- Wenn mehrere Array-Werte ausgewählt wurden, gibt der Befehl die neue Länge des zuletzt aktualisierten Arrays zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.
- SYNTAXERRFehler, wenn eines der JSON-Eingabeargumente keine gültige JSON Zeichenfolge ist.
- NONEXISTENT-Fehler, wenn der Pfad nicht vorhanden ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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"],["a","b"],["c"]]"
```

Eingeschränkte Pfad-Syntax:

```
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

Sucht nach dem ersten Vorkommen eines JSON Skalarwerts in den Arrays am Pfad.

- Fehler außerhalb des Bereichs werden behandelt, indem der Index auf den Anfang und das Ende des Arrays gerundet wird.
- Wenn `start > end`, return -1 (nicht gefunden).

### Syntax

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

- `key` (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom Dokumenttyp. JSON
- `path` (erforderlich) — Ein JSON Pfad.
- `json-scalar` (erforderlich) – Der Skalarwert für die Suche nach . JSONSkalar bezieht sich auf Werte, die keine Objekte oder Arrays sind. Das bedeutet, dass die Werte „Zeichenfolge“, „Zahl“, „Boolescher Wert“ und „Null“ skalare Werte sind.
- `start` (optional) – einschließlich Startindex. Es gilt der Standardwert „0“, falls nicht vorhanden.
- `end` (optional) – Der Endindex (exklusive). Es gilt der Standardwert „0“, wenn keine Angabe gemacht wird, was bedeutet, dass das letzte Element enthalten ist. 0 oder -1 bedeutet, dass das letzte Element enthalten ist.

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von ganzen Zahlen. Jeder Wert ist der Index des übereinstimmenden Elements im Array am Pfad. Der Wert ist -1, falls nicht gefunden.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, der Index des übereinstimmenden Elements oder -1, falls nicht gefunden.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.

### Beispiele

## Erweiterte Pfad-Syntax:

```
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
```

## Eingeschränkte Pfad-Syntax:

```
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

Fügt einen oder mehrere Werte in die Array-Werte beim Pfad vor dem Index ein.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.
- **path** (erforderlich) — Ein JSON Pfad.
- **index** (erforderlich) – Ein Array-Index, vor dem Werte eingefügt werden.
- **json** (erforderlich) — Der JSON Wert, der an das Array angehängt werden soll.

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert ein leeres Array ist, ist der entsprechende Rückgabewert Null.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.

- OUTFBOUNDARIES-Fehler, wenn das Index-Argument außerhalb des gültigen Bereichs liegt.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge des Arrays.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.
- OUTFBOUNDARIES-Fehler, wenn das Index-Argument außerhalb des gültigen Bereichs liegt.

## Beispiele

Erweiterte Pfad-Syntax:

```
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\"]]"
```

Eingeschränkte Pfad-Syntax:

```
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

Ruft die Länge der Array-Werte am Pfad ab.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

## Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die Array-Länge bei jedem Pfad darstellen.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Array von Bulk-Strings. Jedes Element ist ein Schlüsselname im Objekt.
- Ganzzahl, Array-Länge.
- Wenn mehrere Objekte ausgewählt sind, gibt der Befehl die Länge des ersten Arrays zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.
- WRONGTYPE-Fehler, wenn der Pfad nicht vorhanden ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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)

### Eingeschränkte Pfad-Syntax:

```
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

Entfernt und gibt ein Element am Index aus dem Array zurück. Das Öffnen eines leeren Arrays gibt null zurück.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.
- **index** (optional) — Die Position im Array, von der aus das Popping beginnen soll.
  - Ist standardmäßig -1, falls nicht angegeben, was auf das letzte Element verweist.
  - Negativer Wert bedeutet Position vom letzten Element.
  - Außerhalb der Grenzen liegende Indizes werden auf ihre jeweiligen Array-Grenzen gerundet.

## Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Bulk-Strings, die eingeblendete Werte auf jedem Pfad darstellen.
- Wenn ein Wert ein leeres Array ist, ist der entsprechende Rückgabewert Null.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Bulk-Zeichenfolge, die den JSON Pop-up-Wert darstellt.
- Null, wenn das Array leer ist.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Array ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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\"]]"
```

Eingeschränkte Pfad-Syntax:

```
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

Trimmt einen Array am Pfad, der dadurch zum Subarray [Start, Ende] wird (beides inklusive).

- Wenn das Array leer ist, tun Sie nichts, und geben Sie 0 zurück.
- Wenn start < 0 ist, behandeln Sie es als 0.
- Wenn end >= size (Größe des Arrays), behandeln Sie es als size-1.
- Wenn start >= size oder start > end, leeren Sie das Array und geben 0 zurück.

## Syntax

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

- key (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- path (erforderlich) — Ein JSON Pfad.
- start (erforderlich) – Der Startindex, einschließlich.
- end (erforderlich) – Der Endindex, einschließlich.

## Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert ein leeres Array ist, ist der entsprechende Rückgabewert Null.
- Wenn ein Wert kein Array ist, ist der entsprechende Rückgabewert Null.
- OUTFOUBOUNDARIES-Fehler, wenn ein Indexargument außerhalb des gültigen Bereichs liegt.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge des Arrays.
- Null, wenn das Array leer ist.

- **WRONGTYPE-Fehler**, wenn der Wert im Pfad kein Array ist.
- **OUTOFBOUNDARIES-Fehler**, wenn ein Indexargument außerhalb des gültigen Bereichs liegt.

## Beispiele

### Erweiterte Pfad-Syntax:

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

### Eingeschränkte Pfad-Syntax:

```
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

Entfernt die Arrays oder ein Objekt auf dem Pfad.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

### Ergebnis

- Ganzzahl, die Anzahl der entfernten Container.
- Das Löschen eines leeren Arrays oder Objekts führt zu einem entfernten Container.
- Das Löschen eines Nicht-Container-Werts gibt 0 zurück.

## Beispiele

```
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

Meldet Informationen. Unterstützte Unterbefehle sind:

- MEMORY<key>[path] — Meldet die Speichernutzung eines JSON Werts in Byte. Falls nicht angegeben, gilt der Root-Standardwert für den Pfad.
- FIELDS<key>[path] — Meldet die Anzahl der Felder im angegebenen Dokumentpfad. Falls nicht angegeben, gilt der Root-Standardwert für den Pfad. Jeder JSON Wert, der kein Container ist, zählt als ein Feld. Objekte und Arrays zählen rekursiv ein Feld für jeden ihrer enthaltenen Werte. JSON Jeder Containerwert, mit Ausnahme des Root-Containers, zählt als ein zusätzliches Feld.
- HELP— Druckt die Hilfemeldungen des Befehls aus.

## Syntax

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

Hängt vom Unterbefehl ab:

## MEMORY

- Wenn der Pfad eine erweiterte Syntax ist:
  - Gibt ein Array von Ganzzahlen zurück, die die Speichergröße (in Byte) des JSON Werts in jedem Pfad darstellen.
  - Gibt ein leeres Array zurück, wenn der Valkey- oder OSS Redis-Schlüssel nicht existiert.
- Wenn der Pfad eine eingeschränkte Syntax hat:
  - Gibt eine Ganzzahl, die Speichergröße und den JSON Wert in Byte zurück.
  - Gibt Null zurück, wenn der Valkey- oder OSS Redis-Schlüssel nicht existiert.

## FIELDS

- Wenn der Pfad eine erweiterte Syntax ist:
  - Gibt ein Array von Ganzzahlen zurück, die die Anzahl der JSON Wertfelder in jedem Pfad darstellen.
  - Gibt ein leeres Array zurück, wenn der Valkey- oder OSS Redis-Schlüssel nicht existiert.
- Wenn der Pfad eine eingeschränkte Syntax hat:
  - Gibt eine Ganzzahl zurück, die Anzahl der Felder des WertsJSON.
  - Gibt Null zurück, wenn der Valkey- oder OSS Redis-Schlüssel nicht existiert.

HELP— Gibt ein Array von Hilfmeldungen zurück.

## Beispiele

### Erweiterte Pfad-Syntax:

```
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
```

### Eingeschränkte Pfad-Syntax:

```
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

Löscht die JSON Werte am Pfad in einem Dokumentschlüssel. Wenn es sich bei dem Pfad um das Stammverzeichnis handelt, entspricht dies dem Löschen des Schlüssels aus Valkey oder Redis. OSS

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom Dokumenttyp. JSON
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

### Ergebnis

- Anzahl der gelöschten Elemente.
- 0, wenn der Valkey- oder OSS Redis-Schlüssel nicht existiert.
- 0, wenn der JSON Pfad ungültig ist oder nicht existiert.

### Beispiele

#### Erweiterte Pfad-Syntax:

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

#### Eingeschränkte Pfad-Syntax:

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

## JSON.GET

Gibt das serialisierte JSON auf einem oder mehreren Pfaden zurück.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.
- **INDENT/NEWLINE/SPACE**(optional) — Steuert das Format der zurückgegebenen JSON Zeichenfolge, d. h. „pretty print“. Der Standardwert in jedem Fall ist eine leere Zeichenfolge. Die Werte können in jeder Kombination überschrieben werden. Sie können in beliebiger Reihenfolge angegeben werden.
- **NOESCAPE**— Optional, darf aus Gründen der Kompatibilität mit älteren Versionen vorhanden sein und hat keine weiteren Auswirkungen.
- **path** (optional) — Null oder mehr JSON Pfade, standardmäßig das Stammverzeichnis, falls keiner angegeben ist. Die Pfadargumente müssen am Ende platziert werden.

### Ergebnis

Erweiterte Pfad-Syntax:

Wenn ein Pfad angegeben ist:

- Gibt eine serialisierte Zeichenfolge eines Arrays von Werten zurück.
- Wenn kein Wert ausgewählt ist, gibt der Befehl ein leeres Array zurück.

Wenn mehrere Pfade angegeben sind:

- Gibt ein stringifiziertes JSON Objekt zurück, bei dem jeder Pfad ein Schlüssel ist.
- Wenn es gemischte, erweiterte und eingeschränkte Pfadsyntax gibt, entspricht das Ergebnis der erweiterten Syntax.
- Wenn ein Pfad nicht existiert, ist der entsprechende Wert ein leeres Array.

Beispiele

Erweiterte Pfad-Syntax:

```
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
{"\$.firstName":["John\t","\$.lastName":["Smith\t","\$.age":["27\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 $.*
["{},{\ta":1},{\ta":1,\tb":2},1,1,2]"
```

Eingeschränkte Pfad-Syntax:

```
127.0.0.1:6379> JSON.SET k1 .
'{"firstName":"John","lastName":"Smith","age":27,"weight":135.25,"isAlive":true,"address":
```

```

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

```

## JSON.MGET

Wird JSONs am Pfad aus mehreren Dokumentschlüsseln serialisiert. Es gibt Null für einen Schlüssel oder JSON Pfad zurück, der nicht existiert.

### Syntax

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

- **key** (erforderlich) — Ein oder mehrere Valkey- oder OSS Redis-Schlüssel des Dokumenttyps.
- **path** (erforderlich) — Ein JSON Pfad.

### Ergebnis

- Array von Bulk-Strings. Die Größe des Arrays entspricht der Anzahl der Schlüssel im Befehl. Jedes Element des Arrays wird entweder mit (a) dem durch den Pfad angegebenen serialisierten JSON Wert oder (b) mit Null aufgefüllt, wenn der Schlüssel nicht existiert, der Pfad nicht im Dokument existiert oder der Pfad ungültig ist (Syntaxfehler).
- Wenn einer der angegebenen Schlüssel existiert und kein JSON Schlüssel ist, gibt der Befehl einen WRONGTYPE Fehler zurück.

### Beispiele

#### Erweiterte Pfad-Syntax:

```
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\"]
```

### Eingeschränkte Pfad-Syntax:

```
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

Gibt die Zahlenwerte am Pfad durch eine bestimmte Zahl in Schritten an.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.

- path (erforderlich) — Ein JSON Pfad.
- Zahl (erforderlich) – Eine Zahl

## Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Bulk-Strings, die eingeblendete Werte auf jedem Pfad darstellen.
- Wenn ein Wert keine Zahl ist, ist der entsprechende Rückgabewert Null.
- WRONGTYPE-Fehler, wenn die Zahl nicht analysiert werden kann.
- OVERFLOW-Fehler, wenn das Ergebnis außerhalb des IEEE 64-Bit-Double-Bereichs liegt.
- NONEXISTENT wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Bulk-String, der den daraus resultierenden Wert darstellt.
- Wenn mehrere Werte ausgewählt wurden, gibt der Befehl das Ergebnis des zuletzt aktualisierten Arrays zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad keine Zahl ist.
- WRONGTYPE-Fehler, wenn die Zahl nicht analysiert werden kann.
- OVERFLOW-Fehler, wenn das Ergebnis außerhalb des Bereichs von IEEE 64-Bit-Double liegt.
- NONEXISTENT wenn der Dokumentschlüssel nicht vorhanden ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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\":{\"a\":\"a\"},\"b\":{\"a\":\"a\", \"b\":1}, \"c\":{\"a\":\"a\", \"b\":\"b\"}, \"d\":{\"a\":1, \"b\":\"b\", \"c\":3}}"

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}}"

```

### Eingeschränkte Pfad-Syntax:

```
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

Multipliziert die Zahlenwerte am Pfad durch eine bestimmte Zahl.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- **path** (erforderlich) — Ein JSON Pfad.
- **Zahl** (erforderlich) – Eine Zahl

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Bulk-Strings, die den sich daraus ergebenden Wert auf jedem Pfad darstellen.
- Wenn ein Wert keine Zahl ist, ist der entsprechende Rückgabewert Null.
- **WRONGTYPE**-Fehler, wenn die Zahl nicht analysiert werden kann.
- **OVERFLOW**-Fehler, wenn das Ergebnis außerhalb des Bereichs einer 64-Bit-Gleitkommazahl IEEE mit doppelter Genauigkeit liegt.
- **NONEXISTENT** wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Bulk-String, der den daraus resultierenden Wert darstellt.
- Wenn mehrere Werte ausgewählt wurden, gibt der Befehl das Ergebnis des zuletzt aktualisierten Arrays zurück.

- WRONGTYPE-Fehler, wenn der Wert im Pfad keine Zahl ist.
- WRONGTYPE-Fehler, wenn die Zahl nicht analysiert werden kann.
- OVERFLOW-Fehler, wenn das Ergebnis außerhalb des Bereichs einer IEEE 64-Bit-Doppelzahl liegt.
- NONEXISTENT wenn der Dokumentschlüssel nicht vorhanden ist.

## Beispiele

### Erweiterte Pfad-Syntax:

```
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]"
```

## Eingeschränkte Pfad-Syntax:

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

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

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

```

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

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

```

## JSON.OBJLEN

Ruft die Anzahl der Schlüssel in den Objektwerten am Pfad ab.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die Objektlänge in jedem Pfad darstellen.
- Wenn ein Wert kein Objekt ist, ist der entsprechende Rückgabewert null.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, Anzahl der Schlüssel im Objekt.
- Wenn mehrere Objekte ausgewählt sind, gibt der Befehl die Länge des ersten Objekts zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Objekt ist.
- WRONGTYPE-Fehler, wenn der Pfad nicht vorhanden ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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)
```

## Eingeschränkte Pfad-Syntax:

```
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

Ruft Schlüsselnamen in den Objektwerten am Pfad ab.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

## Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Arrays von Bulk-Strings. Jedes Element ist ein Array von Schlüsseln in einem übereinstimmenden Objekt.
- Wenn ein Wert kein Objekt ist, ist der entsprechende Rückgabewert leer.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Array von Bulk-Strings. Jedes Element ist ein Schlüsselname im Objekt.
- Wenn mehrere Objekte ausgewählt sind, gibt der Befehl die Schlüssel des ersten Objekts zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Objekt ist.
- WRONGTYPE-Fehler, wenn der Pfad nicht vorhanden ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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"

## Eingeschränkte Pfad-Syntax:

```
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

Gibt den JSON Wert am angegebenen Pfad im Valkey- oder OSS Redis-Serialisierungsprotokoll () zurück. RESP Wenn der Wert Container ist, ist die Antwort ein RESP Array oder ein verschachteltes Array.

- JSONnull ist der RESP Null-Bulk-String zugeordnet.
- JSONBoolesche Werte werden den jeweiligen Simple Strings zugeordnet. RESP
- Ganzzahlen werden Ganzzahlen zugeordnet. RESP
- IEEEEDoppelte 64-Bit-Gleitkommazahlen werden Bulk Strings zugeordnetRESP.
- JSONZeichenketten werden RESP Bulk Strings zugeordnet.
- JSONArrays werden als RESP Arrays dargestellt, wobei das erste Element die einfache Zeichenfolge [ist, gefolgt von den Elementen des Arrays.
- JSONObjekte werden als RESP Arrays dargestellt, wobei das erste Element die einfache Zeichenfolge {ist, gefolgt von Schlüssel-Wert-Paaren, von denen jedes eine Massenzeichenfolge ist. RESP

## Syntax

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

- **key** (erforderlich) — Ein Valkey- oder Redis-Schlüssel vom DokumenttypOSS. JSON
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

## Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Arrays. Jedes Array-Element repräsentiert die RESP Form des Werts in einem Pfad.
- Leeres Array wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Array, das die RESP Form des Werts im Pfad darstellt.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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"

### Eingeschränkte Pfad-Syntax:

```
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

Legt JSON Werte für den Pfad fest.

Wenn der Pfad ein Objektmitglied aufruft:

- Wenn das übergeordnete Element nicht existiert, gibt der Befehl einen NONEXISTENT Fehler zurück.
- Wenn das übergeordnete Element existiert, aber kein Objekt ist, kehrt der Befehl zurückERROR.
- Wenn das übergeordnete Element existiert und ein Objekt ist:
  - Wenn das Mitglied nicht existiert, wird ein neues Mitglied an das übergeordnete Objekt angehängt, wenn das übergeordnete Objekt das letzte untergeordnete Objekt im Pfad ist. Andernfalls gibt der Befehl einen NONEXISTENT Fehler zurück.
  - Wenn das Mitglied existiert, wird sein Wert durch den JSON Wert ersetzt.

Wenn der Pfad einen Array-Index aufruft:

- Wenn das übergeordnete Element nicht existiert, gibt der Befehl einen NONEXISTENT Fehler zurück.
- Wenn das übergeordnete Element existiert, aber kein Array ist, gibt der Befehl zurückERROR.
- Wenn das übergeordnete Element existiert, der Index jedoch außerhalb der Grenzen liegt, gibt der Befehl einen OUTFOUBOUNDARIES Fehler zurück.
- Wenn das übergeordnete Element existiert und der Index gültig ist, wird das Element durch den neuen JSON Wert ersetzt.

Wenn der Pfad ein Objekt oder ein Array aufruft, wird der Wert (Objekt oder Array) durch den neuen JSON Wert ersetzt.

### Syntax

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

[NX | XX] Wo Sie 0 oder 1 von [NX | XX] Identifikatoren haben können.

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.
- **path** (erforderlich) — Ein JSON Pfad. Für einen neuen Schlüssel muss der JSON Pfad der Stamm „.“ sein.
- **NX** (optional) — Wenn es sich bei dem Pfad um das Stammverzeichnis handelt, legen Sie den Wert nur fest, wenn der Schlüssel nicht existiert. Das heißt, sie fügen ein neues Dokument ein. Wenn der Pfad nicht der Stamm ist, legen Sie den Wert nur fest, wenn der Pfad nicht existiert. Das heißt, Sie fügen einen Wert in das Dokument ein.
- **XX** (optional) — Wenn es sich bei dem Pfad um das Stammverzeichnis handelt, legen Sie den Wert nur fest, wenn der Schlüssel vorhanden ist. Das heißt, Sie ersetzen das vorhandene Dokument. Wenn der Pfad nicht der Stamm ist, legen Sie den Wert nur fest, wenn der Pfad existiert. Das heißt, Sie aktualisieren den vorhandenen Wert.

## Ergebnis

- Einfache Zeichenfolge 'OK' bei Erfolg.
- Null, wenn die NX- oder XX-Bedingungen nicht erfüllt sind.

## Beispiele

### Erweiterte Pfad-Syntax:

```
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]}"
```

### Eingeschränkte Pfad-Syntax:

```
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) OUTFOUBOUNDARIES Array index is out of bounds
```

## JSON.STRAPPEND

Hängt eine Zeichenfolge an die JSON Zeichenketten im Pfad an.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.
- **json\_string** (erforderlich) — Die JSON Darstellung einer Zeichenfolge. Beachten Sie, dass eine JSON Zeichenfolge in Anführungszeichen gesetzt werden muss. Zum Beispiel: ""Beispiel für eine Zeichenfolge""

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die neue Länge des Arrays bei jedem Pfad darstellen.
- Wenn ein Wert im Pfad keine Zeichenfolge ist, ist der entsprechende Rückgabewert Null.
- SYNTAXERRFehler, wenn das Eingabe-JSON-Argument keine gültige JSON Zeichenfolge ist.
- NONEXISTENT-Fehler, wenn der Pfad nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die neue Länge der Zeichenfolge.

- Wenn mehrere Zeichenfolgenwerte ausgewählt wurden, gibt der Befehl die neue Länge der zuletzt aktualisierten Zeichenfolge zurück.
- WRONGTYPE-Fehler, wenn der Wert im Pfad keine Zeichenfolge ist.
- WRONGTYPEFehler, wenn das Eingabe-JSON-Argument keine gültige JSON Zeichenfolge ist.
- NONEXISTENT-Fehler, wenn der Pfad nicht vorhanden ist.

## Beispiele

### Erweiterte Pfad-Syntax:

```
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)

### Eingeschränkte Pfad-Syntax:

```
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

Ruft die Längen der JSON Zeichenkettenwerte im Pfad ab.

### Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen, die die Länge des Zeichenfolgenwerts bei jedem Pfad darstellen.
- Wenn ein Wert keine Zeichenfolge ist, ist der entsprechende Rückgabewert Null.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Ganzzahl, die Länge der Zeichenfolge.
- Wenn mehrere Zeichenfolgenwerte ausgewählt wurden, gibt der Befehl die erste Zeichenfolgenlänge zurück.
- **WRONGTYPE**-Fehler, wenn der Wert im Pfad keine Zeichenfolge ist.
- **NONEXISTENT**-Fehler, wenn der Pfad nicht vorhanden ist.
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.

## Beispiele

### Erweiterte Pfad-Syntax:

```
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)
```

### Eingeschränkte Pfad-Syntax:

```
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

Ein-/Ausblenden der Booleschen Werte zwischen true und false im Pfad.

## Syntax

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

- **key** (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel des JSON Dokumenttyps.
- **path** (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

## Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Ganzzahlen (0 - falsch, 1 - wahr), die den booleschen Wert in jedem Pfad repräsentieren.
- Wenn ein Wert kein Boolescher Wert ist, ist der entsprechende Rückgabewert Null.
- NONEXISTENT wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Zeichenfolge („true“/“false“), die den Booleschen Wert repräsentiert.
- NONEXISTENT wenn der Dokumentschlüssel nicht vorhanden ist.
- WRONGTYPE-Fehler, wenn der Wert im Pfad kein Boolescher Wert ist.

## Beispiele

Erweiterte Pfad-Syntax:

```
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)

### Eingeschränkte Pfad-Syntax:

```
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

Meldet die Art der Werte im jeweiligen Pfad.

### Syntax

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

- key (erforderlich) — Ein Valkey- oder OSS Redis-Schlüssel vom JSON Dokumenttyp.
- path (optional) — Ein JSON Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

### Ergebnis

Wenn der Pfad eine erweiterte Syntax ist:

- Array von Zeichenfolgen, die die Werte in jedem Pfad repräsentieren. Typ {„null“, „boolean“, „string“, „number“, „integer“, „object“ und „array“}.
- Wenn ein Pfad nicht existiert, ist der entsprechende Ausgabewert null.
- Leeres Array wenn der Dokumentschlüssel nicht vorhanden ist.

Wenn der Pfad eine eingeschränkte Syntax hat:

- Zeichenfolge, Typ des Werts
- Null, wenn der Dokumentschlüssel nicht vorhanden ist.
- Null, wenn der JSON Pfad ungültig ist oder nicht existiert.

## Beispiele

Erweiterte Pfad-Syntax:

```
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
```

Eingeschränkte Pfad-Syntax:

```
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
```

## Verschlagworten Sie Ihre Ressourcen ElastiCache

Um Ihnen bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen zu helfen, können Sie jeder Ressource Ihre eigenen Metadaten in Form von Tags zuweisen. Mithilfe von Tags können Sie Ihre AWS Ressourcen auf unterschiedliche Weise kategorisieren, z. B. nach Zweck, Eigentümer oder Umgebung. Dies ist nützlich, wenn Sie viele Ressourcen desselben Typs haben — In diesem Fall können Sie schnell bestimmte Ressourcen basierend auf den zugewiesenen Tags (Markierungen) bestimmen. In diesem Thema werden Tags (Markierungen) und deren Erstellung beschrieben.

### Warning

Als bewährte Vorgehensweise empfehlen wir Ihnen, keine sensiblen Daten in Ihre Tags (Markierungen) aufzunehmen.

## Grundlagen zu Tags (Markierungen)

Ein Tag ist eine Bezeichnung, die Sie einer AWS Ressource zuweisen. Jeder Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert, beides können Sie bestimmen. Mithilfe von Tags können Sie Ihre AWS Ressourcen auf unterschiedliche Weise kategorisieren, z. B. nach Zweck oder Eigentümer. Sie könnten beispielsweise eine Reihe von Stichwörtern für die ElastiCache Cluster Ihres Kontos definieren, mit deren Hilfe Sie den Besitzer und die Benutzergruppe jeder Instanz nachverfolgen können.

Wir empfehlen die Verwendung von Tag (Markierung)-Schlüsseln, die die Anforderungen der jeweiligen Ressourcentypen erfüllen. Die Verwendung einheitlicher Tag-Schlüssel vereinfacht

das Verwalten der -Ressourcen. Sie können die Ressourcen auf Grundlage der hinzugefügten Tags (Markierungen) filtern und danach suchen. Weitere Informationen zum Implementieren einer effektiven Ressourcen-Markierungs-Strategie finden Sie im [-Whitepaper AWS Bewährte Methoden zur Markierung](#).

Tags haben keine semantische Bedeutung ElastiCache und werden ausschließlich als Zeichenfolge interpretiert. Außerdem werden Tags (Markierungen) nicht automatisch Ihren Ressourcen zugewiesen. Sie können Tag (Markierung)-Schlüssel und -Werte bearbeiten und Tags (Markierungen) jederzeit von einer Ressource entfernen. Sie können den Wert eines Tags auf null setzen. Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben. Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht. Wenn Sie einer Replikationsgruppe Tags hinzufügen oder löschen, werden außerdem die Tags aller Knoten in dieser Replikationsgruppe hinzugefügt oder entfernt.

Sie können mit Tags arbeiten, indem Sie die AWS Management Console AWS CLI, und die ElastiCache API verwenden.

Wenn Sie Tags verwenden IAM, können Sie steuern, welche Benutzer in Ihrem AWS Konto berechtigt sind, Tags zu erstellen, zu bearbeiten oder zu löschen. Weitere Informationen finden Sie unter [Berechtigungen auf Ressourcenebene](#).

## Ressourcen, die markiert werden können

Sie können die meisten ElastiCache Ressourcen, die bereits in Ihrem Konto vorhanden sind, taggen. In der Tabelle unten werden die Ressourcen aufgeführt, die das Markieren unterstützen. Wenn Sie den verwenden AWS Management Console, können Sie mithilfe des [Tag-Editors](#) Tags auf Ressourcen anwenden. Auf einigen Ressourcenbildschirmen können Sie beim Erstellen der Ressource Tags für diese Ressource angeben, z. B. ein Tag mit dem Schlüssel „Name“ und einem von Ihnen angegebenen Wert. In den meisten Fällen wendet die Konsole Tags (Markierungen) direkt nach dem Erstellen der Ressource an und nicht während des Erstellens. Die Konsole kann Ressourcen nach dem Name-Tag organisieren, aber dieses Tag hat für den ElastiCache Dienst keine semantische Bedeutung.

Zudem können Sie mit einigen Aktionen zur Ressourcenerstellung Tags beim Erstellen einer Ressource angeben. Wenn Tags (Markierungen) nicht während der Ressourcenerstellung angewendet werden können, wird die Ressourcenerstellung rückgängig gemacht. Auf diese Weise werden Ressourcen entweder mit Tags (Markierungen) oder überhaupt nicht erstellt und keine

Ressourcen verbleiben ohne Tags (Markierungen). Indem Sie Ressourcen zum Erstellungszeitpunkt markieren, müssen Sie anschließend keine benutzerdefinierten Skripts ausführen.

Wenn Sie Amazon ElastiCache API, The oder An AWS SDK verwenden AWS CLI, können Sie den Tags Parameter für die entsprechende ElastiCache API Aktion verwenden, um Tags anzuwenden. Diese sind:

- CreateServerlessCache
- CreateCacheCluster
- CreateReplicationGroup
- CopyServerlessCacheSnapshot
- CopySnapshot
- CreateCacheParameterGroup
- CreateCacheSecurityGroup
- CreateCacheSubnetGroup
- CreateServerlessCacheSnapshot
- CreateSnapshot
- CreateUserGroup
- CreateUser
- PurchaseReservedCacheNodesOffering

In der folgenden Tabelle werden die ElastiCache Ressourcen beschrieben, die markiert werden können, und die Ressourcen, die bei der Erstellung mit dem ElastiCache API AWS CLI, dem oder einem markiert werden können AWS SDK.

Unterstützung für ElastiCache das Markieren von Ressourcen

Unterstützt Tags (Markierungen)	Unterstützt Markierung bei der Erstellung
Ja	Ja
Ja	Ja

Unterstützt Tags (Markierungen)	Unterstützt Markierung bei der Erstellung
Ja	Ja

 Note

Globale Datenspeicher können nicht mit Tags versehen werden.

Sie können in Ihren IAM Richtlinien tagbasierte Berechtigungen auf Ressourcenebene auf ElastiCache API Aktionen anwenden, die das Markieren bei der Erstellung unterstützen, um eine detaillierte Kontrolle über die Benutzer und Gruppen zu erreichen, die Ressourcen bei der Erstellung taggen können. Ihre Ressourcen sind vor der Erstellung ordnungsgemäß gesichert – Tags, die sofort auf Ihre Ressourcen angewendet werden. Daher sind alle tagbasierten Berechtigungen auf Ressourcenebene, welche die Verwendung von Ressourcen steuern, sofort wirksam. Ihre

Ressourcen können nachverfolgt und genauer erfasst werden. Sie können das Markieren neuer Ressourcen gewährleisten und steuern, welche Tag (Markierung)-Schlüssel und Werte für Ihre Ressourcen festgelegt sind.

Weitere Informationen finden Sie unter [Beispiele für das Taggen von Ressourcen](#).

Weitere Informationen zur Markierung von Ressourcen für die Fakturierung finden Sie unter [Überwachung von Kosten mit Kostenzuordnungs-Tags](#).

## Markieren von Caches und Snapshots

Die folgenden Regeln gelten für das Markieren im Rahmen von Anforderungsvorgängen:

- `CreateReplicationGroup`:
  - Wenn `--primary-cluster-id` und `--tags`-Parameter in der Anforderung enthalten sind, werden die Anforderungs-Tags der Replikationsgruppe hinzugefügt und an alle Cache-Cluster in der Replikationsgruppe weitergegeben. Wenn der primäre Cache-Cluster über vorhandene Tags verfügt, werden diese mit den Anforderungs-Tags überschrieben, um konsistente Tags auf allen Knoten zu haben.

Wenn keine Anforderungs-Tags vorhanden sind, werden die primären Cache-Cluster-Tags der Replikationsgruppe hinzugefügt und an alle Cache-Cluster weitergegeben.

- Wenn `--snapshot-name` oder `--serverless-cache-snapshot-name` angegeben wird:

Wenn die Anforderung Tags enthält, wird die Replikationsgruppe nur mit diesen Tags markiert. Wenn die Anforderung keine Tags enthält, werden der Replikationsgruppe die Snapshot-Tags hinzugefügt.

- Wenn `--global-replication-group-id` geliefert wird:

Wenn die Anforderung Tags enthält, werden die Anforderungs-Tags der Replikationsgruppe hinzugefügt und an alle Cache-Cluster weitergegeben.

- `CreateCacheCluster` :

- Wenn `--replication-group-id` geliefert wird:

Wenn die Anforderung Tags enthält, wird der Cache-Cluster nur mit diesen Tags getaggt. Wenn die Anforderung keine Tags enthält, erbt der Cache-Cluster die Replikationsgruppen-Tags anstelle der Tags des primären Cache-Clusters.

- Wenn `--snapshot-name` geliefert wird:

Wenn die Anforderung Tags enthält, wird der Cache-Cluster nur mit diesen Tags getaggt. Wenn die Anforderung keine Tags enthält, werden die Snapshot-Tags dem Cache-Cluster hinzugefügt.

- `CreateServerlessCache` :
  - Wenn die Anforderung Tags enthält, werden dem Serverless-Cache nur die Anforderungs-Tags hinzugefügt.
- `CreateSnapshot` :
  - Wenn `--replication-group-id` geliefert wird:

Wenn die Anforderung Tags enthält, werden nur die Anfrage-Tags zum Snapshot hinzugefügt. Wenn die Anforderung keine Tags enthält, werden die Replikationsgruppen-Tags zum Snapshot hinzugefügt.

- Wenn `--cache-cluster-id` geliefert wird:

Wenn die Anforderung Tags enthält, werden nur die Anfrage-Tags zum Snapshot hinzugefügt. Wenn die Anforderung keine Tags enthält, werden die Cache-Cluster-Tags zum Snapshot hinzugefügt.

- Für automatische Snapshots:

Tags werden von den Replikationsgruppen-Tags weitergegeben.

- `CreateServerlessCacheSnapshot` :
  - Wenn die Anforderung Tags enthält, werden dem Serverless-Cache-Snapshot nur die Anforderungs-Tags hinzugefügt.
- `CopySnapshot` :
  - Wenn die Anforderung Tags enthält, werden nur die Request-Tags zum Snapshot hinzugefügt. Wenn die Anforderung keine Tags enthält, werden die Quell-Snapshot-Tags zum kopierten Snapshot hinzugefügt.
- `CopyServerlessCacheSnapshot` :
  - Wenn die Anforderung Tags enthält, werden dem Serverless-Cache-Snapshot nur die Anforderungs-Tags hinzugefügt.
- `AddTagsToResource` und `RemoveTagsFromResource`
  - Tags werden zur Replikationsgruppe hinzugefügt/entfernt und die Aktion wird an alle Cluster in der Replikationsgruppe weitergegeben.

**Note**

AddTagsToResource und RemoveTagsFromResource kann nicht für Standardparameter und Sicherheitsgruppen verwendet werden.

- IncreaseReplicaCount und ModifyReplicationGroupShardConfiguration:
  - Für alle neuen Cluster, die der Replikationsgruppe hinzugefügt wurden, werden dieselben Tags wie für die Replikationsgruppe angewendet.

## Tag-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Maximale Anzahl von Tags (Markierungen) pro Ressource: 50
- Jeder Tag (Markierung) muss für jede Ressource eindeutig sein. Jeder Tag (Markierung) kann nur einen Wert haben.
- Maximale Schlüssellänge — 128 Unicode-Zeichen in UTF -8.
- Maximale Wertlänge — 256 Unicode-Zeichen in UTF -8.
- Obwohl ElastiCache jedes Zeichen in seinen Tags zulässig ist, können andere Dienste restriktiv sein. In allen Diensten sind folgende Zeichen zulässig: Buchstaben, Zahlen und Leerzeichen, die in UTF -8 dargestellt werden können, sowie die folgenden Zeichen: + - = . \_ : / @
- Bei Tag-Schlüsseln und -Werten wird zwischen Groß- und Kleinschreibung unterschieden.
- Das `aws :` Präfix ist für AWS die Verwendung reserviert. Wenn der Tag (Markierung) über einen Tag (Markierung)-Schlüssel mit diesem Präfix verfügt, können Sie den Schlüssel oder Wert des Tags (Markierung) nicht bearbeiten oder löschen. Tags (Markierungen) mit dem Präfix `aws :` werden nicht als Ihre Tags (Markierungen) pro Ressourcenlimit angerechnet.

Sie können Ressourcen nicht allein auf Grundlage ihrer Tags (Markierungen) beenden, anhalten oder löschen. Sie müssen den Ressourcenbezeichner angeben. Um Snapshots zu löschen, die Sie mit dem Tag (Markierung)-Schlüssel `DeleteMe` markiert haben, müssen Sie die `DeleteSnapshot`-Aktion mit den Ressourcenbezeichnern der Snapshots verwenden, z. B. `snap-1234567890abcdef0`.

Weitere Informationen zu ElastiCache Ressourcen, die Sie taggen können, finden Sie unter [Ressourcen, die markiert werden können](#).

## Beispiele für das Taggen von Ressourcen

- Einen serverlosen Cache mithilfe von Tags erstellen. In diesem Beispiel wird Memcached als Engine verwendet.

```
aws elasticache create-serverless-cache \
 --serverless-cache-name CacheName \
 --engine memcached \
 --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Hinzufügen von Tags zu einem Serverless-Cache

```
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"
```

- Hinzufügen von Tags zu einer Replikationsgruppe.

```
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"
```

- Erstellen eines Cache-Clusters mithilfe von Tags.

```
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"
```

- Erstellen eines Cache-Clusters mithilfe von Tags. In diesem Beispiel wird Redis als Engine verwendet.

```
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"
```

```
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Erstellen eines Serverless-Snapshots mit Tags. In diesem Beispiel wird Memcached als Engine verwendet.

```
aws elasticache create-serverless-cache-snapshot \
--serverless-cache-name testing-tags \
--serverless-cache-snapshot-name bkp-testing-tags-scs \
--tags Key="work",Value="foo"
```

- Erstellen eines Snapshots mit Tags.

Snapshots sind derzeit nur für Redis verfügbar. Wenn Sie in diesem Fall Tags auf Anfrage hinzufügen, erhält der Snapshot nur die Anfrage-Tags, selbst wenn die Replikationsgruppe Tags enthält.

```
aws elasticache create-snapshot \
--replication-group-id testing-tags \
--snapshot-name bkp-testing-tags-rg \
--tags Key="work",Value="foo"
```

## Beispiele für Tag-basierte Zugriffssteuerungsrichtlinien

1. AddTagsToResourceAktionen für einen Cluster sind nur zulässig, wenn der Cluster das Tag Project= hat. XYZ

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "elasticache:AddTagsToResource",
 "Resource": [
 "arn:aws:elasticache:*:*:cluster:*"
],
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/Project": "XYZ"
 }
 }
 }
]
}
```

```

 }
]
}

```

2. Zulassen von RemoveTagsFromResource-Aktion von einer Replikationsgruppe, wenn sie die Tags „Project“ und „Service“ enthält und Schlüssel sich von „Project“ und „Service“ unterscheiden.

```

{
 "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"
]
 }
 }
 }
]
}

```

3. Zulassen von AddTagsToResource zu einer Ressource nur dann, wenn sich Tags von „Project“ und „Service“ unterscheiden.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "elasticache:AddTagsToResource",
 "Resource": [
 "arn:aws:elasticache:*:*:*:*"
]
 }
]
}

```

```

],
 "Condition": {
 "ForAnyValue:StringNotEqualsIgnoreCase": {
 "aws:TagKeys": [
 "Service",
 "Project"
]
 }
 }
]
}

```

4. Verweigerung CreateReplicationGroup-Aktion, wenn die Anfrage Tag Project=Foo enthält.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Deny",
 "Action": "elasticache:CreateReplicationGroup",
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/Project": "Foo"
 }
 }
 }
]
}

```

5. Die CopySnapshot Aktion wird verweigert, wenn der Quell-Snapshot das Tag Project= XYZ und das Anforderungs-Tag Service=Elasticache hat.

```

{
 "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. Verweigern der Aktion `CreateCacheCluster`, wenn das Anforderungs-Tag `Project` fehlt oder nicht `Dev`, `QA` oder `Prod` entspricht.

```

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

Weitere Informationen zu Bedingungsschlüsseln finden Sie unter [Verwenden von Bedingungsschlüssel](#).

## Überwachung von Kosten mit Kostenzuordnungs-Tags

Wenn Sie Ihren Ressourcen in Amazon Kostenzuordnungs-Tags hinzufügen ElastiCache, können Sie die Kosten verfolgen, indem Sie die Ausgaben auf Ihren Rechnungen nach Ressourcen-Tag-Werten gruppieren.

Ein ElastiCache Kostenzuweisungs-Tag ist ein Schlüssel-Wert-Paar, das Sie definieren und einer Ressource zuordnen. ElastiCache Bei Schlüsseln und Werten werden Groß-/Kleinschreibung berücksichtigt. Sie können einen Tag-Schlüssel verwenden, um eine Kategorie zu definieren, und der Tag-Wert kann ein Element in dieser Kategorie sein. So könnten Sie beispielsweise den Tag-Schlüssel `CostCenter` und den Tag-Wert `10010` definieren, um anzugeben, dass die Ressource der Kostenstelle 10010 zugewiesen ist. Sie können mit Tags auch Ressourcen kennzeichnen, die zu Test- oder Produktionszwecken verwendet werden, indem Sie einen Schlüssel wie z. B. `Environment` und Werte wie z. B. `test` oder `production` verwenden. Wir empfehlen, einheitliche

Tag-Schlüssel zu verwenden, um die mit Ihren Ressourcen verknüpften Kosten einfacher verfolgen zu können.

Verwenden Sie Kostenzuordnungs-Tags, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Melden Sie sich dazu an, um Ihre AWS Kontorechnung mit den Tag-Schlüsselwerten zu erhalten. Um dann die Kosten kombinierter Ressourcen anzuzeigen, organisieren Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag-Schlüsselwerten. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können.

Sie können auch Tags miteinander kombinieren, um Kosten detaillierter zu verfolgen. Um beispielsweise Ihre Kosten für Services nach Region zu verfolgen, könnten Sie die Tag-Schlüssel `Service` und `Region` verwenden. Für eine Ressource lauten die Werte möglicherweise `ElastiCache` und `Asia Pacific (Singapore)` und für eine andere Ressource lauten sie `ElastiCache` und `Europe (Frankfurt)`. Sie können dann Ihre ElastiCache Gesamtkosten nach Regionen aufgeschlüsselt sehen. Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#) im AWS Billing -Benutzerhandbuch.

Sie können ElastiCache selbst entworfenen Clustern Tags für die ElastiCache Kostenzuweisung hinzufügen. Wenn Sie ein Tag hinzufügen, auflisten, ändern, kopieren oder entfernen, wird die Operation nur auf die angegebenen Cluster angewendet.

### Eigenschaften von ElastiCache Kostenzuweisungs-Tags

- Kostenzuweisungs-Tags werden auf ElastiCache Ressourcen angewendet, die in CLI und API Operationen als angegeben sindARN. Der Ressourcentyp ist ein „Cluster“.

BeispielARN: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- Der Tag-Schlüssel ist der erforderliche Name des Tags. Der Zeichenfolgenwert kann aus 1 bis 128 Unicode-Zeichen bestehen. Ihm darf kein `aws:` als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen, Unterstriche (`_`), Punkte (`.`), Doppelpunkte (`:`), Backslashes (`\`), Gleichheitszeichen (`=`), Pluszeichen (`+`), Trennstriche (`-`) oder At-Zeichen (`@`) enthalten.

- Der Tag-Wert ist der optionale Wert des Tags. Der Zeichenfolgenwert kann aus 1 bis 256 Unicode-Zeichen bestehen. Ihm darf kein `aws:` als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen, Unterstriche (`_`), Punkte (`.`), Doppelpunkte (`:`), Backslashes (`\`), Gleichheitszeichen (`=`), Pluszeichen (`+`), Trennstriche (`-`) oder At-Zeichen (`@`) enthalten.
- Eine ElastiCache Ressource kann maximal 50 Tags haben.
- Die Werte innerhalb eines Tag-Satzes müssen nicht eindeutig sein. Beispiel: In einem Tag-Satz könnten die Schlüssel `Service` und `Application` beide den Wert `ElastiCache` besitzen.

AWS wendet Ihren Tags keine semantische Bedeutung an. Tags werden ausschließlich als Zeichenketten interpretiert. AWS setzt nicht automatisch irgendwelche Tags für eine ElastiCache Ressource.

## Verwaltung Ihrer Kostenzuweisungs-Tags mithilfe der AWS CLI

Sie können die verwenden, AWS CLI um Kostenzuordnungs-Tags hinzuzufügen, zu ändern oder zu entfernen.

Kostenzuweisungs-Tags werden auf ElastiCache Cluster angewendet. Der Cluster, der markiert werden soll, wird mit einem ARN (Amazon-Ressourcennamen) angegeben.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

### Themen

- [Auflisten von Tags mit dem AWS CLI](#)
- [Hinzufügen von Tags mit dem AWS CLI](#)
- [Ändern von Tags mit dem AWS CLI](#)
- [Entfernen von Tags mit dem AWS CLI](#)

## Auflisten von Tags mit dem AWS CLI

Sie können die verwenden AWS CLI , um Tags für eine vorhandene ElastiCache Ressource aufzulisten, indem Sie den [list-tags-for-resource](#) Operation.

Der folgende Code verwendet die AWS CLI , um die Tags auf dem Memcached-Cluster `my-cluster` in der Region `us-west-2` aufzulisten.

Für Linux, macOS oder Unix:

```
aws elasticache list-tags-for-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Für Windows:

```
aws elasticache list-tags-for-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Der folgende Code verwendet die AWS CLI , um die Tags auf dem Valkey- oder OSS Redis-Knoten `my-cluster-001` im `my-cluster` Cluster in der Region `us-west-2` aufzulisten.

Für Linux, macOS oder Unix:

```
aws elasticache list-tags-for-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Für Windows:

```
aws elasticache list-tags-for-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus und besteht aus einer Liste aller Tags für die Ressource.

```
{
 "TagList": [
 {
 "Value": "10110",
 "Key": "CostCenter"
 },
 {
 "Value": "EC2",
 "Key": "Service"
 }
]
}
```

```
}
```

Wenn die Ressource keine Tags enthält, ist die Ausgabe leer. TagList

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

Weitere Informationen finden Sie im AWS CLI für ElastiCache [list-tags-for-resource](#).

## Hinzufügen von Tags mit dem AWS CLI

Sie können das verwenden AWS CLI , um einer vorhandenen ElastiCache Ressource Tags hinzuzufügen, indem Sie den [add-tags-to-resource](#) CLIOperation. Wenn das Tag in der Ressource nicht vorhanden ist, werden Schlüssel und Wert zur Ressource hinzugefügt. Wenn der Schlüssel in der Ressource bereits vorhanden ist, wird der diesem Schlüssel zugeordnete Wert auf den neuen Wert aktualisiert.

Der folgende Code verwendet die AWS CLI , um die Schlüssel Service und Region mit den Werten `elasticache` `us-west-2` jeweils zum Knoten `my-cluster-001` im Cluster `my-cluster` in der Region `us-west-2` hinzuzufügen.

### Memcached

Für Linux, macOS oder 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
```

Für 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

Für Linux, macOS oder 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
```

Für 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
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus und besteht aus einer Liste aller Tags für die Ressource nach der Operation.

```
{
 "TagList": [
 {
 "Value": "elasticache",
 "Key": "Service"
 },
 {
 "Value": "us-west-2",
 "Key": "Region"
 }
]
}
```

Weitere Informationen finden Sie im für AWS CLI ElastiCache [add-tags-to-resource](#).

Sie können den auch verwenden AWS CLI , um einem Cluster Tags hinzuzufügen, wenn Sie mithilfe des Vorgangs einen neuen Cluster erstellen [create-cache-cluster](#). Sie können keine Tags hinzufügen, wenn Sie einen Cluster mit der ElastiCache Managementkonsole erstellen. Nachdem der Cluster erstellt wurde, können Sie über die Konsole Tags zum Cluster hinzufügen.

## Ändern von Tags mit dem AWS CLI

Sie können den verwenden AWS CLI , um die Tags auf einem ElastiCache Cluster zu ändern.

Ändern Sie Tags wie folgt:

- Verwenden Sie [add-tags-to-resource](#) um entweder ein neues Tag und einen neuen Wert hinzuzufügen oder den Wert zu ändern, der einem vorhandenen Tag zugeordnet ist.
- Verwenden Sie [remove-tags-from-resource](#) um bestimmte Tags aus der Ressource zu entfernen.

zum Entfernen von Tags aus dem Cluster.

## Entfernen von Tags mit dem AWS CLI

Sie können den verwenden AWS CLI , um Tags aus einem vorhandenen ElastiCache (Memcached-) Cluster zu entfernen, indem Sie den [remove-tags-from-resource](#) Vorgang.

Für Memcached verwendet der folgende Code die, AWS CLI um die Tags mit den Schlüsseln Service und Region aus dem Knoten `my-cluster-001` im Cluster `my-cluster` in der Region `us-west-2` zu entfernen.

Für Linux, macOS oder Unix:

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
 --tag-keys PM Service
```

Für Windows:

```
aws elasticache remove-tags-from-resource ^
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
 --tag-keys PM Service
```

Für Redis verwendet der folgende Code die, AWS CLI um die Tags mit den Schlüsseln Service und Region vom Knoten `my-cluster-001` im Cluster `my-cluster` in der Region `us-west-2` zu entfernen.

Für Linux, macOS oder Unix:

```
aws elasticache remove-tags-from-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 \
 --tag-keys PM Service
```

Für Windows:

```
aws elasticache remove-tags-from-resource ^
--resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001 ^
--tag-keys PM Service
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus und besteht aus einer Liste aller Tags für die Ressource nach der Operation.

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

Weitere Informationen finden Sie im für AWS CLI ElastiCache [remove-tags-from-resource](#).

## Verwaltung Ihrer Kostenzuweisungs-Tags mithilfe der ElastiCache API

Sie können die verwenden, ElastiCache API um Kostenzuordnungs-Tags hinzuzufügen, zu ändern oder zu entfernen.

Kostenzuweisungs-Tags werden ElastiCache für Memcached-Cluster angewendet. Der Cluster, der markiert werden soll, wird mit einem ARN (Amazon-Ressourcennamen) angegeben.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Themen

- [Auflisten von Tags unter Verwendung von ElastiCache API](#)
- [Hinzufügen von Tags mit dem ElastiCache API](#)
- [Ändern von Tags mit dem ElastiCache API](#)
- [Entfernen von Tags mit dem ElastiCache API](#)

### Auflisten von Tags unter Verwendung von ElastiCache API

Sie können die verwenden ElastiCache API, um Tags für eine vorhandene Ressource aufzulisten, indem Sie [ListTagsForResource](#) Operation.

Für Memcached verwendet der folgende Code die, ElastiCache API um die Tags auf der Ressource `my-cluster` in der Region `us-west-2` aufzulisten.

```
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>
```

Für Redis verwendet der folgende Code die, ElastiCache API um die Tags auf der Ressource `my-cluster-001` in der Region `us-west-2` aufzulisten.

```
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>
```

## Hinzufügen von Tags mit dem ElastiCache API

Sie können das verwenden ElastiCache API, um einem vorhandenen ElastiCache Cluster Tags hinzuzufügen, indem Sie [AddTagsToResource](#) Vorgang. Wenn das Tag in der Ressource nicht vorhanden ist, werden Schlüssel und Wert zur Ressource hinzugefügt. Wenn der Schlüssel in der Ressource bereits vorhanden ist, wird der diesem Schlüssel zugeordnete Wert auf den neuen Wert aktualisiert.

Der folgende Code verwendet die ElastiCache API, um die Schlüssel `Service` und `Region` mit den Werten `elasticache us-west-2` bzw. hinzuzufügen. Bei Memcached wird dies auf die Ressource `my-cluster` angewendet. Für Redis wird dies auf die Ressource `my-cluster-001` in der Region `US-West-2` angewendet.

### 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>
```

Weitere Informationen finden Sie unter [AddTagsToResource](#) in der ElastiCache API Amazon-Referenz.

## Ändern von Tags mit dem ElastiCache API

Sie können den verwenden ElastiCache API, um die Tags auf einem ElastiCache Cluster zu ändern.

Den Wert eines Tags ändern:

- Verwenden Sie [AddTagsToResource](#) Operation, um entweder ein neues Tag und einen neuen Wert hinzuzufügen oder den Wert eines vorhandenen Tags zu ändern.
- Verwenden Sie [RemoveTagsFromResource](#) um Tags aus der Ressource zu entfernen.

Die Ausgabe beider Operationen ist eine Liste der Tags und deren Werte für die angegebene Ressource.

Verwenden Sie [RemoveTagsFromResource](#) um Tags aus der Ressource zu entfernen.

## Entfernen von Tags mit dem ElastiCache API

Sie können das verwenden ElastiCache API, um Tags aus einem vorhandenen ElastiCache (Memcached-) Cluster zu entfernen, indem Sie [RemoveTagsFromResource](#) Vorgang.

Der folgende Code verwendet die ElastiCache API, um die Tags mit den Schlüsseln `Service` und `Region` vom Knoten `my-cluster-001` im Cluster `my-cluster` in der Region `us-west-2` zu entfernen.

```
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>
```

## Verwenden des Amazon ElastiCache Well-Architected-Objektivs

In diesem Abschnitt wird Amazon ElastiCache Well-Architected Lens beschrieben, eine Sammlung von Entwurfsprinzipien und Anleitungen für die Gestaltung gut ElastiCache strukturierter Workloads.

- Die ElastiCache Linse ist eine Ergänzung zum [AWS Well-Architected Framework](#).
- Jede Säule hat eine Reihe von Fragen, die helfen sollen, die Diskussion rund um eine ElastiCache Architektur zu beginnen.
  - Jede Frage enthält eine Reihe von führenden Methoden sowie deren Ergebnisse für die Berichterstattung.
    - Erforderlich – notwendig vor dem Wechsel in die Produktion (sofern kein hohes Risiko besteht)
    - Am besten – bestmöglicher Zustand für einen Kunden
    - Gut – unsere Empfehlungen für den Kunden (sofern kein mittleres Risiko besteht)
- Well-Architected-Terminologie
  - [Komponente](#) — Code, Konfiguration und AWS Ressourcen, die zusammen eine Anforderung erfüllen. Komponenten interagieren mit anderen Komponenten und entsprechen in Microservice-Architekturen häufig einem Service.

- [Arbeitslast](#) — Eine Reihe von Komponenten, die zusammen einen geschäftlichen Nutzen bieten. Beispiele für Workloads sind Marketing-Websites, E-Commerce-Websites, das Backend für eine mobile App, Analyseplattformen usw.

### Note

Dieses Handbuch wurde nicht aktualisiert und enthält nun auch Informationen zum ElastiCache serverlosen Caching und zur neuen Valkey-Engine.

## Themen

- [Säule Operational Excellence von Amazon ElastiCache Well-Architected Lens](#)
- [Sicherheitssäule von Amazon ElastiCache Well-Architected Lens](#)
- [Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens](#)
- [Säule der Leistung und Effizienz von Amazon ElastiCache Well-Architected Lens](#)
- [Säule zur ElastiCache Kostenoptimierung bei Amazon Well-Architected Lens](#)

## Säule Operational Excellence von Amazon ElastiCache Well-Architected Lens

Die Säule „Operational Excellence“ konzentriert sich auf den Betrieb und die Überwachung von Systemen, um einen Mehrwert für das Unternehmen zu schaffen, sowie auf die kontinuierliche Verbesserung von Prozessen und Verfahren. Zu den wichtigsten Themen gehören die Automatisierung von Änderungen, die Reaktion auf Ereignisse und die Definition von Standards für die Verwaltung des täglichen Betriebs.

## Themen

- [OE 1: Wie verstehen Sie Warnmeldungen und Ereignisse, die von Ihrem ElastiCache Cluster ausgelöst werden, und wie reagieren Sie darauf?](#)
- [OE 2: Wann und wie skalieren Sie Ihre vorhandenen Cluster? ElastiCache](#)
- [OE 3: Wie verwalten Sie Ihre ElastiCache Cluster-Ressourcen und pflegen Ihren Cluster up-to-date?](#)
- [OE 4: Wie verwalten Sie die Verbindungen der Kunden zu Ihren ElastiCache Clustern?](#)
- [OE 5: Wie werden ElastiCache Komponenten für einen Workload bereitgestellt?](#)

- [OE 6: Wie planen und minimieren Sie Ausfälle?](#)
- [OE 7: Wie behebt man Valkey- oder OSS Redis-Engine-Ereignisse?](#)

## OE 1: Wie verstehen Sie Warnmeldungen und Ereignisse, die von Ihrem ElastiCache Cluster ausgelöst werden, und wie reagieren Sie darauf?

Einführung auf Fragenebene: Wenn Sie ElastiCache Cluster betreiben, können Sie optional Benachrichtigungen und Warnmeldungen erhalten, wenn bestimmte Ereignisse eintreten. ElastiCache protokolliert standardmäßig [Ereignisse](#), die sich auf Ihre Ressourcen beziehen, z. B. Failover, Knotenaustausch, Skalierungsvorgänge, geplante Wartungsarbeiten und mehr. Jedes Ereignis enthält das Datum und die Uhrzeit, den Quellnamen und den Quelltyp sowie eine Beschreibung.

Vorteil auf Fragenebene: Wenn Sie in der Lage sind, die zugrundeliegenden Ursachen für die Ereignisse, die von Ihrem Cluster generierte Warnmeldungen auslösen, zu verstehen und zu verwalten, können Sie effektiver arbeiten und angemessen auf Ereignisse reagieren.

- [Erforderlich] Überprüfen Sie die Ereignisse, die von ElastiCache auf der ElastiCache Konsole (nach Auswahl Ihrer Region) oder mithilfe des Befehls [describe-events](#) von [Amazon Command Line Interface](#) (AWS CLI) und dem generiert wurden. [ElastiCache API](#) Konfigurieren Sie ElastiCache das Senden von Benachrichtigungen für wichtige Cluster-Ereignisse mithilfe von Amazon Simple Notification Service (AmazonSNS). Wenn Sie Amazon SNS mit Ihren Clustern verwenden, können Sie bei ElastiCache Ereignissen programmgesteuert Maßnahmen ergreifen.
- Es gibt zwei große Ereigniskategorien: aktuelle und geplante Ereignisse. Die Liste der aktuellen Ereignisse umfasst: Erstellung und Löschung von Ressourcen, Skalierungsvorgänge, Failover, Neustart des Knotens, erstellter Snapshot, Änderung der Cluster-Parameter, Erneuerung des CA-Zertifikats, Fehlerereignisse (Cluster-Bereitstellungsfehler - VPC oder -, Skalierungsfehler ENI - - und Snapshot-Fehler). ENI Die Liste der geplanten Ereignisse umfasst: Knoten, dessen Austausch während des Wartungsfensters geplant ist, und Knotenaustausch, der verschoben wurde.
- Auch wenn Sie auf einige dieser Ereignisse möglicherweise nicht sofort reagieren müssen, ist es wichtig, sich zunächst alle Fehlerereignisse anzusehen:
  - ElastiCache:AddCacheNodeFailed
  - ElastiCache:CacheClusterProvisioningFailed
  - ElastiCache:CacheClusterScalingFailed
  - ElastiCache:CacheNodesRebooted

- ElastiCache: SnapshotFailed (Nur Valkey oder Redis) OSS
- [Ressourcen]:
  - [ElastiCache SNS Amazon-Benachrichtigungen verwalten](#)
  - [Event-Benachrichtigungen und Amazon SNS](#)
- [Am besten] Nutzen Sie AWS Produkt- und Servicefunktionen wie SNS Lambda Functions, um Reaktionen auf Ereignisse zu automatisieren. Halten Sie sich an bewährte Methoden, indem Sie kleine, häufige, umkehrbare Änderungen als Code vornehmen, um Ihre Operationen im Laufe der Zeit weiterzuentwickeln. Sie sollten CloudWatch Amazon-Metriken verwenden, um Ihre Cluster zu überwachen.

[Ressourcen]: [Überwachen Sie ElastiCache \(RedisOSS\) \(Cluster-Modus deaktiviert\), lesen Sie Replica-Endpunkte mit AWS Lambda, Amazon Route 53 und Amazon SNS für einen Anwendungsfall](#), der Lambda und verwendet. SNS

## OE 2: Wann und wie skalieren Sie Ihre vorhandenen Cluster? ElastiCache

Einführung auf Fragenebene: Die richtige Größe Ihres ElastiCache Clusters ist ein Balanceakt, der bei jeder Änderung der zugrunde liegenden Workload-Typen bewertet werden muss. Ihr Ziel ist es, mit der richtigen Größe für Ihren Workload zu arbeiten.

Vorteil auf Fragenebene: Eine Überlastung Ihrer Ressourcen kann zu einer erhöhten Latenz und einer insgesamt verringerten Leistung führen. Eine Unterauslastung kann andererseits zu einer Überbereitstellung von Ressourcen bei nicht optimaler Kostenoptimierung führen. Durch die richtige Dimensionierung Ihrer Umgebungen können Sie ein Gleichgewicht zwischen Leistungseffizienz und Kostenoptimierung erreichen. Um eine Über- oder Unterauslastung Ihrer Ressourcen zu beheben, ElastiCache können Sie in zwei Dimensionen skalieren. Sie können vertikal skalieren, indem Sie die Knotenkapazität erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie Knoten hinzufügen und entfernen.

- [Erforderlich] CPU und Netzwerküberlastung auf primären Knoten sollten behoben werden, indem die Lesevorgänge ausgelagert und auf Replikatknoten umgeleitet werden. Verwenden Sie Replikatknoten für Lesevorgänge, um die Auslastung des Primärknotens zu reduzieren. Dies kann in Ihrer Valkey- oder OSS Redis-Client-Bibliothek konfiguriert werden, indem Sie eine Verbindung zum ElastiCache Reader-Endpunkt herstellen, wenn der Clustermodus deaktiviert ist, oder indem Sie den Befehl für den Clustermodus aktiviert verwenden. READONLY

[Ressourcen]:

- [Verbindungsendpunkte finden in ElastiCache](#)
- [Richtige Clustergröße](#)
- [READONLYBefehl](#)
- [Erforderlich] Überwachen Sie die Nutzung kritischer Clusterressourcen wie CPU Arbeitsspeicher und Netzwerk. Die Auslastung dieser spezifischen Clusterressourcen muss nachverfolgt werden, um eine Entscheidung für eine Skalierung und die Art des Skalierungsvorgangs treffen zu können. Wenn der Clustermodus ElastiCache (RedisOSS) deaktiviert ist, können Primär- und Replikatknoten vertikal skaliert werden. Replikatknoten können auch horizontal von 0 auf 5 Knoten skaliert werden. Wenn der Clustermodus aktiviert ist, gilt dasselbe für jeden Shard Ihres Clusters. Darüber hinaus können Sie die Anzahl der Shards erhöhen oder reduzieren.

[Ressourcen]:

- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) unter Verwendung von Amazon CloudWatch](#)
- [Skalierung von ElastiCache \(Redis OSS -\) Clustern](#)
- [Skalierung ElastiCache für Memcached-Cluster](#)
- [Am besten] Wenn Sie Trends im Zeitverlauf überwachen, können Sie Workload-Änderungen erkennen, die bei punktueller Überwachung unbemerkt bleiben würden. Um längerfristige Trends zu erkennen, verwenden Sie CloudWatch Metriken, um nach längeren Zeiträumen zu suchen. Die Erkenntnisse aus der Beobachtung von CloudWatch Kennzahlen über längere Zeiträume sollten als Grundlage für Ihre Prognose zur Auslastung der Cluster-Ressourcen dienen. CloudWatch Datenpunkte und Metriken sind für bis zu 455 Tage verfügbar.

[Ressourcen]:

- [Überwachung ElastiCache \(RedisOSS\) mit Metriken CloudWatch](#)
- [Überwachung von Memcached mit Metriken CloudWatch](#)
- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) unter Verwendung von Amazon CloudWatch](#)
- [Optimal] Wenn Ihre ElastiCache Ressourcen damit erstellt werden, empfiehlt CloudFormation es sich, Änderungen mithilfe von CloudFormation Vorlagen vorzunehmen, um die betriebliche Konsistenz zu wahren und unverwaltete Konfigurationsänderungen und Stack-Drifts zu vermeiden.

[Ressourcen]:

- [ElastiCache Referenz zum Ressourcentyp für CloudFormation](#)

- [Am besten] Automatisieren Sie Ihre Skalierungsvorgänge mithilfe von Cluster-Betriebsdaten und definieren Sie Schwellenwerte für CloudWatch die Einrichtung von Alarmen. Verwenden Sie CloudWatch Events und Simple Notification Service (SNS), um Lambda-Funktionen auszulösen und auszuführen und Ihre Cluster automatisch ElastiCache API zu skalieren. Ein Beispiel wäre, Ihrem Cluster einen Shard hinzuzufügen, wenn die Metrik `EngineCPUUtilization` über einen längeren Zeitraum 80 % erreicht. Eine weitere Option wäre die Verwendung von `DatabaseMemoryUsedPercentages` für einen speicherbasierten Schwellenwert.

[Ressourcen]:

- [Amazon CloudWatch Alarms verwenden](#)
- [Was sind CloudWatch Amazon-Events?](#)
- [Verwendung AWS Lambda mit Amazon Simple Notification Service](#)
- [ElastiCacheAPIReferenz](#)

### OE 3: Wie verwalten Sie Ihre ElastiCache Cluster-Ressourcen und pflegen Ihren Cluster up-to-date?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, ist es wichtig, dass Sie in der Lage sind, all Ihre Ressourcen zu lokalisieren und zu identifizieren. ElastiCache Bei der Einführung neuer Anwendungsfunktionen müssen Sie für eine Symmetrie der Cluster-Versionen in all Ihren ElastiCache Umgebungstypen sorgen: Entwicklung, Testen und Produktion. Mithilfe von Ressourcenattributen können Sie Umgebungen für unterschiedliche betriebliche Ziele trennen, z. B. bei der Einführung neuer Funktionen und der Aktivierung neuer Sicherheitsmechanismen.

Vorteil auf Fragenebene: Die Trennung Ihrer Entwicklungs-, Test- und Produktionsumgebungen ist die beste betriebliche Methode. Es ist auch eine bewährte Methode, dass auf Ihren Clustern und Knoten in allen Umgebungen die neuesten Softwarepatches installiert werden, wobei wohlverstandene und dokumentierte Prozesse verwendet werden. Durch die Nutzung systemeigener ElastiCache Funktionen kann sich Ihr Entwicklungsteam auf die Erreichung der Geschäftsziele konzentrieren und nicht auf die ElastiCache Wartung.

- [Am besten] Verwenden Sie die neueste verfügbare Engine-Version und installieren Sie die Self-Service-Updates, sobald sie verfügbar sind. ElastiCache aktualisiert die zugrunde liegende Infrastruktur automatisch während des angegebenen Wartungsfensters des Clusters. Die in Ihren Clustern laufenden Knoten werden jedoch über Self-Service-Updates aktualisiert. Es gibt zwei Arten von Updates: Sicherheitspatches oder kleinere Software-Updates. Stellen Sie sicher, dass

Sie den Unterschied zwischen den Patch-Typen verstehen und wissen, wann diese angewendet werden.

[Ressourcen]:

- [Self-Service-Updates bei Amazon ElastiCache](#)
- [Hilfeseite für Amazon ElastiCache Managed Maintenance and Service Updates](#)
- [Am besten] Organisieren Sie Ihre ElastiCache Ressourcen mithilfe von Tags. Verwenden Sie Tags für Replikationsgruppen und nicht für einzelne Knoten. Sie können Tags so konfigurieren, dass sie angezeigt werden, wenn Sie Ressourcen abfragen, und Sie können Tags verwenden, um Suchen durchzuführen und Filter anzuwenden. Sie sollten Ressourcengruppen verwenden, um auf einfache Weise Sammlungen von Ressourcen zu erstellen und zu verwalten, die gemeinsame Tag-Sätze verwenden.

[Ressourcen]:

- [Bewährte Methoden zum Taggen](#)
- [ElastiCache Referenz zum Ressourcentyp für CloudFormation](#)
- [Parametergruppen](#)

## OE 4: Wie verwalten Sie die Verbindungen der Kunden zu Ihren ElastiCache Clustern?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, müssen Sie verstehen, wie Ihre Kunden mit dem ElastiCache Cluster kommunizieren, um die betrieblichen Aspekte Ihrer Anwendung (z. B. Reaktionszeiten) zu verwalten.

Vorteil auf Fragenebene: Durch die Auswahl des am besten geeigneten Verbindungsmechanismus wird sichergestellt, dass Ihre Anwendung nicht aufgrund von Verbindungsfehlern, wie Timeouts, unterbrochen wird.

- [Erforderlich] Trennen Sie Lese- und Schreibvorgänge und stellen Sie eine Verbindung mit dem Replikatknoten her, um Lesevorgänge auszuführen. Beachten Sie jedoch, dass Sie, wenn Sie die Schreibvorgänge von den Lesevorgängen trennen, aufgrund der asynchronen Natur der Valkey- und Redis-Replikation die Fähigkeit verlieren, einen Schlüssel unmittelbar nach dem Schreiben zu lesen. OSS Der WAIT Befehl kann genutzt werden, um die Datensicherheit in der Praxis zu verbessern und Replikat dazu zu zwingen, Schreibvorgänge zu bestätigen, bevor sie den Clients antworten, was wiederum zu Leistungseinbußen führt. Die Verwendung von Replikatknoten für Lesevorgänge kann in Ihrer ElastiCache (Redis-OSS) Clientbibliothek konfiguriert werden, indem der ElastiCache Reader-Endpunkt für den Clustermodus deaktiviert

ist. Wenn der Clustermodus aktiviert ist, verwenden Sie den Befehl ElastiCache (RedisOSS).  
READONLY Für viele der ElastiCache (Redis-OSS) Clientbibliotheken READONLY ist ElastiCache (RedisOSS) standardmäßig oder über eine Konfigurationseinstellung implementiert.

[Ressourcen]:

- [Verbindungsendpunkte finden in ElastiCache](#)
- [READONLY](#)
- [Erforderlich] Verwenden Sie Verbindungspooling. Das Herstellen einer TCP Verbindung ist sowohl auf der Client- als auch auf der Serverseite mit CPU Zeitaufwand verbunden, und das Pooling ermöglicht es Ihnen, die Verbindung wiederzuverwenden. TCP

Um den Verbindungsaufwand zu reduzieren, sollten Sie Verbindungspooling verwenden. Mit einem Pool von Verbindungen kann Ihre Anwendung Verbindungen „nach Belieben“ wiederverwenden und freigeben, ohne dass Kosten für den Verbindungsaufbau anfallen. Sie können das Verbindungspooling über Ihre ElastiCache (Redis-OSS) Clientbibliothek (sofern unterstützt) implementieren, wobei ein Framework für Ihre Anwendungsumgebung verfügbar ist, oder es von Grund auf neu erstellen.

- [Am besten] Stellen Sie sicher, dass das Socket-Timeout des Clients auf mindestens eine Sekunde eingestellt ist (im Vergleich zur typischen Standardeinstellung „Keine“ bei verschiedenen Clients).
  - Wenn Sie den Timeout-Wert zu niedrig einstellen, kann dies zu möglichen Timeouts führen, wenn die Serverlast hoch ist. Eine zu hohe Einstellung kann dazu führen, dass Ihre Anwendung lange braucht, um Verbindungsprobleme zu erkennen.
  - Kontrollieren Sie das Volumen neuer Verbindungen, indem Sie Verbindungspooling in Ihrer Client-Anwendung implementieren. Dies reduziert die Latenz und die CPU Auslastung, die zum Öffnen und Schließen von Verbindungen und zum Ausführen eines TLS Handshakes erforderlich sind, falls dies auf dem TLS Cluster aktiviert ist.

[Ressourcen]: [Konfiguration ElastiCache \(RedisOSS\) für höhere Verfügbarkeit](#)

- [Gut] Der Einsatz von Pipelining (sofern Ihre Anwendungsfälle dies zulassen) kann die Leistung erheblich steigern.
  - Mit Pipelining reduzieren Sie die RTT Round-Trip-Zeit () zwischen Ihren Anwendungsclients und dem Cluster. Neue Anfragen können verarbeitet werden, auch wenn der Client die vorherigen Antworten noch nicht gelesen hat.
  - Mit Pipelining können Sie mehrere Befehle an den Server senden, ohne auf Antworten/ Bestätigungen warten zu müssen. Der Nachteil von Pipelining ist, dass, wenn Sie irgendwann

alle Antworten in Massen abrufen, möglicherweise ein Fehler aufgetreten ist, den Sie erst am Ende erkennen können.

- Implementieren Sie Methoden, um Anfragen erneut zu versuchen, wenn ein Fehler zurückgegeben wird, der die fehlerhafte Anfrage auslöst.

[Ressourcen]: [Pipelining](#)

## OE 5: Wie werden ElastiCache Komponenten für einen Workload bereitgestellt?

Einführung auf Fragenebene: ElastiCache Umgebungen können manuell über die AWS Konsole oder programmgesteuert über APIs, Toolkits usw. bereitgestellt werden. CLI Bewährte Methoden für Operational Excellence sehen vor, Bereitstellungen möglichst mithilfe von Code zu automatisieren. Darüber hinaus können ElastiCache Cluster entweder nach Arbeitslast isoliert oder zur Kostenoptimierung kombiniert werden.

Vorteil auf Frageebene: Durch die Auswahl des für Ihre ElastiCache Umgebungen am besten geeigneten Bereitstellungsmechanismus kann Operation Excellence im Laufe der Zeit verbessert werden. Es wird empfohlen, Operationen möglichst als Code auszuführen, um menschliche Fehler zu minimieren und die Wiederholbarkeit, Flexibilität und Reaktionszeit auf Ereignisse zu erhöhen.

Wenn Sie die Anforderungen an die Workload-Isolierung verstehen, können Sie entscheiden, ob Sie pro Workload dedizierte ElastiCache Umgebungen einrichten oder mehrere Workloads zu einzelnen Clustern oder Kombinationen davon kombinieren möchten. Die Nachteile zu verstehen, kann dazu beitragen, ein Gleichgewicht zwischen Operational Excellence und Kostenoptimierung zu finden

- [Erforderlich] Machen Sie sich mit den verfügbaren Bereitstellungsoptionen vertraut und automatisieren Sie diese Verfahren ElastiCache, wann immer dies möglich ist. Zu den möglichen Automatisierungsmöglichkeiten gehören CloudFormationSDK, AWS CLI/und APIs.

[Ressourcen]:

- [Referenz zum ElastiCache Amazon-Ressourcentyp](#)
- [elasticache](#)
- [ElastiCache APIAmazon-Referenz](#)
- [Erforderlich] Ermitteln Sie für alle Workloads den erforderlichen Grad der Clusterisolierung.
  - [Am besten]: Hohe Isolierung – eine 1:1-Zuordnung zwischen Workload und Cluster. Ermöglicht die feinste Kontrolle über den Zugriff, die Größe, Skalierung und Verwaltung von ElastiCache Ressourcen auf Workload-Basis.

- [Besser]: Mittlere Isolierung – M:1 ist zweckmäßig isoliert, aber möglicherweise von mehreren Workloads gemeinsam genutzt (z. B. ein Cluster, der für das Zwischenspeichern von Workloads vorgesehen ist, und ein anderer, der für Messaging dediziert ist).
- [Gut]: Niedrige Isolierung – M:1 für alle Zwecke, vollständig gemeinsam genutzt. Empfohlen für Workloads, bei denen gemeinsamer Zugriff akzeptabel ist.

## OE 6: Wie planen und minimieren Sie Ausfälle?

Einführung auf Fragenebene: Operational Excellence umfasst die Antizipation von Ausfällen durch regelmäßige „Pre-Mortem“-Übungen zur Identifizierung potenzieller Fehlerquellen, sodass sie behoben oder eingedämmt werden können. ElastiCache bietet ein FailoverAPI, das zu Testzwecken simulierte Knotenausfälle ermöglicht.

Vorteil auf Fragenebene: Wenn Sie Ausfallszenarien vorab testen, können Sie herausfinden, wie sie sich auf Ihren Workload auswirken. Dies ermöglicht ein sicheres Testen der Reaktionsverfahren und ihrer Wirksamkeit und macht Ihr Team mit deren Ausführung vertraut.

[Erforderlich] Führen Sie regelmäßig Failover-Tests in Entwickler-/Testkonten durch. [TestFailover](#)

## OE 7: Wie behebt man Valkey- oder OSS Redis-Engine-Ereignisse?

Einführung auf Fragenebene: Operational Excellence erfordert die Fähigkeit, sowohl Informationen auf Service- als auch auf Engine-Ebene zu untersuchen, um den Zustand und den Status Ihrer Cluster zu analysieren. ElastiCache kann Valkey- oder OSS Redis-Engine-Protokolle sowohl an Amazon als auch an Amazon Kinesis CloudWatch Data Firehose senden.

Vorteil auf Fragenebene: Durch die Aktivierung von Valkey- oder OSS Redis-Engine-Protokollen auf ElastiCache Clustern erhalten Sie Einblicke in Ereignisse, die sich auf den Zustand und die Leistung von Clustern auswirken. Valkey- oder OSS Redis-Engine-Logs liefern Daten direkt von der Engine, die nicht über den Ereignismechanismus verfügbar sind. ElastiCache Durch sorgfältige Beobachtung sowohl der ElastiCache Ereignisse (siehe oben in OE-1) als auch der Engine-Logs ist es möglich, bei der Fehlerbehebung eine Reihenfolge der Ereignisse sowohl aus ElastiCache Service- als auch aus Engine-Perspektive zu bestimmen.

- [Erforderlich] Stellen Sie sicher, dass die Protokollierungsfunktion der OSS Redis-Engine aktiviert ist. Diese Funktion ist ab ElastiCache (RedisOSS) 6.2 und neuer verfügbar. Dieser Vorgang kann während der Clustererstellung oder durch Änderung des Clusters nach der Erstellung erfolgen.

- Stellen Sie fest, ob Amazon CloudWatch Logs oder Amazon Kinesis Data Firehose das geeignete Ziel für OSS Redis-Engine-Protokolle ist.
- Wählen Sie ein geeignetes Zielprotokoll in CloudWatch oder Kinesis Data Firehose aus, um die Protokolle beizubehalten. Wenn Sie mehrere Cluster haben, sollten Sie für jeden Cluster ein anderes Zielprotokoll verwenden, da Daten bei der Fehlerbehebung so besser isoliert werden können.

[Ressourcen]:

- Protokollzustellung: [Protokollzustellung](#)
- Ziele für die Protokollierung: [Amazon CloudWatch Logs](#)
- Einführung in Amazon CloudWatch Logs: [Was ist Amazon CloudWatch Logs?](#)
- Einführung in Amazon Kinesis Data Firehose: [Was ist Amazon Kinesis Data Firehose?](#)
- [Am besten] Wenn Sie Amazon CloudWatch Logs verwenden, sollten Sie Amazon CloudWatch Logs Insights nutzen, um das Valkey- oder OSS Redis-Engine-Protokoll nach wichtigen Informationen abzufragen.

Erstellen Sie als Beispiel eine Abfrage für die CloudWatch Protokollgruppe, die die Valkey- oder OSS Redis-Engine-Protokolle enthält, die Ereignisse mit dem Zeichen "WARNING" zurückgeben, wie zum Beispiel: LogLevel

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Ressourcen]: [Analysieren von Protokolldaten mit CloudWatch](#) Logs Insights

## Sicherheitssäule von Amazon ElastiCache Well-Architected Lens

Die Säule der Sicherheit konzentriert sich auf den Schutz von Informationen und Systemen. Zu den wichtigsten Themen gehören die Vertraulichkeit und Integrität von Daten, die Ermittlung und Verwaltung von Berechtigungen mithilfe der Berechtigungsverwaltung, der Schutz von Systemen und die Einrichtung von Kontrollen zur Erkennung von Sicherheitsereignissen.

Themen

- [SEC1: Welche Schritte ergreifen Sie, um den autorisierten Zugriff auf Daten zu ElastiCache kontrollieren?](#)

- [SEC2: Benötigen Ihre Anwendungen zusätzliche Autorisierungen, die ElastiCache über netzwerkbasierete Kontrollen hinausgehen?](#)
- [SEC3: Besteht das Risiko, dass Befehle versehentlich ausgeführt werden, was zu Datenverlust oder -ausfällen führt?](#)
- [SEC4: Wie stellen Sie die Datenverschlüsselung im Ruhezustand sicher ElastiCache](#)
- [SEC5: Wie verschlüsselt man Daten bei der Übertragung mit? ElastiCache](#)
- [SEC6: Wie schränken Sie den Zugriff auf Ressourcen der Kontrollebene ein?](#)
- [SEC7: Wie erkennen Sie Sicherheitsereignisse und reagieren darauf?](#)

## SEC1: Welche Schritte ergreifen Sie, um den autorisierten Zugriff auf Daten zu ElastiCache kontrollieren?

Einführung auf Fragenebene: Alle ElastiCache Cluster sind für den Zugriff von Amazon Elastic Compute Cloud-Instances in serverlosen Funktionen (AWS Lambda) oder Containern (Amazon Elastic Container Service) konzipiert. VPC Das am häufigsten anzutreffende Szenario ist der Zugriff auf einen ElastiCache Cluster von einer Amazon Elastic Compute Cloud-Instanz innerhalb derselben Amazon Virtual Private Cloud (Amazon Virtual Private Cloud). Bevor Sie von einer EC2 Amazon-Instance aus eine Verbindung zu einem Cluster herstellen können, müssen Sie die EC2 Amazon-Instance autorisieren, auf den Cluster zuzugreifen. Um auf einen ElastiCache Cluster zuzugreifenVPC, der in einem ausgeführt wird, müssen Sie dem Cluster Netzwerkzugang gewähren.

Vorteil auf Frageebene: Der Netzwerkzugang in den Cluster wird über Sicherheitsgruppen gesteuert. VPC Eine Sicherheitsgruppe fungiert als virtuelle Firewall für Ihre EC2 Amazon-Instances, um den eingehenden und ausgehenden Datenverkehr zu kontrollieren. Eingehende Regeln steuern den eingehenden Datenverkehr zu Ihrer Instance und ausgehende Regeln steuern den ausgehenden Datenverkehr von Ihrer Instance. Im Fall von ElastiCache, wenn ein Cluster gestartet wird, muss eine Sicherheitsgruppe zugeordnet werden. Dadurch wird sichergestellt, dass Regeln für ein- und ausgehenden Datenverkehr für alle Knoten gelten, aus denen der Cluster besteht. Darüber hinaus ElastiCache ist es so konfiguriert, dass es ausschließlich in privaten Subnetzen bereitgestellt wird, sodass auf sie nur über das VPC private Netzwerk von zugegriffen werden kann.

- [Erforderlich] Die Ihrem Cluster zugeordnete Sicherheitsgruppe steuert den Netzwerkeingang und den Zugriff auf den Cluster. Standardmäßig sind für eine Sicherheitsgruppe keine Regeln für eingehenden Datenverkehr definiert und daher auch kein Eingangspfad zu. ElastiCache Um dies zu aktivieren, konfigurieren Sie eine Regel für eingehende Nachrichten für die Sicherheitsgruppe, die Quell-IP-Adresse/den Quellbereich, den TCP Typ des Datenverkehrs und den Port für Ihren

ElastiCache Cluster (z. B. Standardport 6379 für ElastiCache (Redis)). OSS Es ist zwar möglich, eine sehr breite Palette von Eingangsquellen zuzulassen, z. B. alle Ressourcen innerhalb einer VPC (0.0.0.0/0), es wird jedoch empfohlen, die Regeln für eingehenden Datenverkehr so detailliert wie möglich zu definieren, z. B. nur den eingehenden Zugriff auf Valkey- oder OSS Redis-Clients zu autorisieren, die auf Amazon-Amazon-Instances laufen, die einer bestimmten Sicherheitsgruppe zugeordnet sind. EC2

[Ressourcen]:

- [Subnetze und Subnetzgruppen](#)
- [Zugriff auf Ihren Cluster oder die Replikationsgruppe](#)
- [Steuern des Datenverkehrs zu Ressourcen mithilfe von Sicherheitsgruppen](#)
- [Sicherheitsgruppen von Amazon Elastic Compute Cloud für Linux-Instances](#)
- [Erforderlich]AWS Identity and Access Management Richtlinien können Funktionen zugewiesen werden, die ihnen den Zugriff auf Daten ermöglichen. AWS Lambda ElastiCache Um diese Funktion zu aktivieren, erstellen Sie eine IAM Ausführungsrolle mit der `AWSLambdaVPCAccessExecutionRole` entsprechenden Berechtigung und weisen Sie die Rolle dann der AWS Lambda Funktion zu.

[Ressourcen]: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einem AmazonVPC: [Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einem Amazon VPC](#)

SEC2: Benötigen Ihre Anwendungen zusätzliche Autorisierungen, die ElastiCache über netzwerkbasierte Kontrollen hinausgehen?

Einführung auf Fragenebene: In Szenarien, in denen es notwendig ist, den Zugriff auf ElastiCache (Redis-OSS) Cluster auf individueller Client-Ebene einzuschränken oder zu kontrollieren, wird empfohlen, sich über den Befehl (Redis) zu authentifizieren. ElastiCache OSS AUTH ElastiCache (RedisOSS) -Authentifizierungstoken mit optionaler Benutzer- und Benutzergruppenverwaltung ermöglichen es ElastiCache (RedisOSS), ein Passwort anzufordern, bevor Clients Befehle und Zugriffsschlüssel ausführen können, wodurch die Sicherheit der Datenebene verbessert wird.

Vorteil auf Frageebene: Um die Sicherheit Ihrer Daten zu gewährleisten, bietet ElastiCache (RedisOSS) Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten. Dazu gehört auch, dass Clients eine rollenbasierte Zugriffskontrolle (RBAC) oder ein AUTH Token (Passwort)

verwenden müssen AUTH, um eine Verbindung herzustellen, bevor autorisierte Befehle ausgeführt werden. ElastiCache

- [Am besten] Definieren Sie für ElastiCache (RedisOSS) 6.x und höher Authentifizierungs- und Autorisierungskontrollen, indem Sie Benutzergruppen, Benutzer und Zugriffszeichenfolgen definieren. Weisen Sie Benutzer Benutzergruppen zu und weisen Sie Benutzergruppen dann Clustern zu. Zur Verwendung muss RBAC es bei der Clustererstellung ausgewählt und die Verschlüsselung bei der Übertragung aktiviert sein. Stellen Sie sicher, dass Sie einen Valkey- oder OSS Redis-Client verwenden, der diese Funktion unterstützt, um die Vorteile nutzen TLS zu können. RBAC

[Ressourcen]:

- [Bewerben Sie RBAC sich bei einer Replikationsgruppe für ElastiCache \(Redis\) OSS](#)
- [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#)
- [ACL](#)
- [Unterstützte ElastiCache \(RedisOSS\) Versionen](#)
- [Am besten] Für ElastiCache (Redis-OSS) Versionen vor 6.x empfiehlt es sich, neben der Festlegung eines starken Tokens/Passworts und der Einhaltung einer strengen Passwortrichtlinie für ElastiCache (RedisOSS) auch das Wechseln des AUTH Passworts/Tokens zu verwenden. ElastiCache kann bis zu zwei (2) Authentifizierungstoken gleichzeitig verwalten. Sie können den Cluster auch so ändern, dass explizit die Verwendung von Authentifizierungstoken erforderlich ist.

[Ressourcen]: [Ändern des AUTH Tokens auf einem vorhandenen ElastiCache \(Redis-OSS\) Cluster](#)

SEC3: Besteht das Risiko, dass Befehle versehentlich ausgeführt werden, was zu Datenverlust oder -ausfällen führt?

Einführung auf Fragenebene: Es gibt eine Reihe von Valkey- oder OSS Redis-Befehlen, die sich negativ auf den Betrieb auswirken können, wenn sie versehentlich oder von böswilligen Akteuren ausgeführt werden. Diese Befehle können im Hinblick auf die Leistung und Datensicherheit unbeabsichtigte Folgen haben. Beispielsweise kann ein Entwickler den FLUSHALL Befehl routinemäßig in einer Entwicklungsumgebung aufrufen und aufgrund eines Fehlers versehentlich versuchen, diesen Befehl auf einem Produktionssystem aufzurufen, was zu versehentlichem Datenverlust führt.

Vorteil auf Fragenebene: Ab ElastiCache (RedisOSS) 5.0.3 haben Sie die Möglichkeit, bestimmte Befehle umzubenennen, die Ihre Arbeitslast stören könnten. Durch das Umbenennen der Befehle kann verhindert werden, dass diese versehentlich auf dem Cluster ausgeführt werden.

- [Erforderlich]

[Ressourcen]:

- [ElastiCache \(RedisOSS\) Version 5.0.3 \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [Änderungen an den Parametern von OSS Redis 5.0.3](#)
- [Redis-Sicherheit OSS](#)

## SEC4: Wie stellen Sie die Datenverschlüsselung im Ruhezustand sicher ElastiCache

Einführung auf Fragenebene: Obwohl es sich bei ElastiCache (RedisOSS) um einen In-Memory-Datenspeicher handelt, ist es im Rahmen der Standardoperationen des Clusters möglich, alle Daten zu verschlüsseln, die möglicherweise dauerhaft (im Speicher) gespeichert werden. Dazu gehören sowohl geplante als auch manuelle Backups, die in Amazon S3 geschrieben wurden, sowie Daten, die aufgrund von Synchronisierungs- und Swap-Vorgängen auf dem Festplattenspeicher gespeichert wurden. Instance-Typen der M6g- und R6g-Familien verfügen außerdem über eine ständig aktive In-Memory-Verschlüsselung.

Vorteil auf Fragenebene: ElastiCache (RedisOSS) bietet optionale Verschlüsselung im Ruhezustand, um die Datensicherheit zu erhöhen.

- [Erforderlich] Die Verschlüsselung im Ruhezustand kann auf einem ElastiCache Cluster (Replikationsgruppe) nur aktiviert werden, wenn dieser erstellt wurde. Ein vorhandener Cluster kann nicht geändert werden, um mit der Verschlüsselung von Daten im Ruhezustand zu beginnen. Stellt standardmäßig die Schlüssel bereit und verwaltet ElastiCache sie, die bei der Verschlüsselung im Ruhezustand verwendet werden.

[Ressourcen]:

- [Einschränkungen bei der Verschlüsselung im Ruhezustand](#)
- [Aktivieren der Verschlüsselung im Ruhezustand](#)
- [Am besten] Nutzen Sie EC2 Amazon-Instance-Typen, die Daten verschlüsseln, während sie sich im Arbeitsspeicher befinden (z. B. M6g oder R6g). Wenn möglich, sollten Sie erwägen, eigene Schlüssel für die Verschlüsselung im Ruhezustand zu verwalten. In Umgebungen mit strengerer Datensicherheit kann AWS Key Management Service (KMS) zur Selbstverwaltung der

Kundenhauptschlüssel () verwendet werden. CMK Durch die ElastiCache Integration mit sind Sie in der Lage AWS Key Management Service, die Schlüssel zu erstellen, zu besitzen und zu verwalten, die für die Verschlüsselung ruhender Daten für Ihren ElastiCache (Redis-OSS) Cluster verwendet werden.

[Ressourcen]:

- [Verwenden Sie vom Kunden verwaltete Schlüssel von AWS Key Management Service](#)
- [AWS Schlüsselverwaltungsdienst](#)
- [AWS KMSKonzepte](#)

## SEC5: Wie verschlüsselt man Daten bei der Übertragung mit? ElastiCache

Einführung auf Fragenebene: Es ist eine gängige Anforderung, zu verhindern, dass Daten während der Übertragung beschädigt werden. Dabei handelt es sich um Daten innerhalb von Komponenten eines verteilten Systems sowie zwischen Anwendungsclients und Clusterknoten. ElastiCache (RedisOSS) unterstützt diese Anforderung, indem es die Verschlüsselung von Daten ermöglicht, die zwischen Clients und Clustern sowie zwischen den Clusterknoten selbst übertragen werden. Instance-Typen der M6g- und R6g-Familien verfügen außerdem über eine ständig aktive In-Memory-Verschlüsselung.

Vorteil auf Frageebene: Die Verschlüsselung ElastiCache während der Übertragung durch Amazon ist eine optionale Funktion, mit der Sie die Sicherheit Ihrer Daten an den anfälligsten Stellen erhöhen können, wenn sie von einem Ort zum anderen übertragen werden.

- [Erforderlich] Die Verschlüsselung bei der Übertragung kann bei der Erstellung nur auf einem ElastiCache (Redis-OSS) Cluster (Replikationsgruppe) aktiviert werden. Bitte beachten Sie, dass die Implementierung der Verschlüsselung bei der Übertragung aufgrund der zusätzlichen Verarbeitung, die für die Ver-/Entschlüsselung von Daten erforderlich ist, Auswirkungen auf die Leistung hat. Um die Auswirkungen zu verstehen, wird empfohlen, Ihren Workload vor und nach der Aktivierung zu vergleichen. encryption-in-transit

[Ressourcen]:

- [Übersicht über die Verschlüsselung bei der Übertragung](#)

## SEC6: Wie schränken Sie den Zugriff auf Ressourcen der Kontrollebene ein?

Einführung auf Fragenebene: IAM Richtlinien und Bereitstellung ARN detaillierter Zugriffskontrollen für ElastiCache (RedisOSS), sodass die Erstellung, Änderung und Löschung von ElastiCache (Redis-) Clustern genauer kontrolliert werden kann. OSS

Vorteil auf Fragenebene: Die Verwaltung von ElastiCache Amazon-Ressourcen wie Replikationsgruppen, Knoten usw. kann auf AWS Konten beschränkt werden, die über spezifische Berechtigungen auf der Grundlage von IAM Richtlinien verfügen, wodurch die Sicherheit und Zuverlässigkeit der Ressourcen verbessert wird.

- [Erforderlich] Verwalten Sie den Zugriff auf ElastiCache Amazon-Ressourcen, indem Sie AWS Benutzern bestimmte AWS Identity and Access Management Richtlinien zuweisen, sodass Sie genauer kontrollieren können, welche Konten welche Aktionen auf Clustern ausführen können.

[Ressourcen]:

- [Überblick über die Verwaltung der Zugriffsberechtigungen für Ihre Ressourcen ElastiCache](#)
- [Verwendung identitätsbasierter Richtlinien \(IAMRichtlinien\) für Amazon ElastiCache](#)

## SEC7: Wie erkennen Sie Sicherheitsereignisse und reagieren darauf?

Einführung auf Fragenebene: ElastiCache Exportiert bei der Bereitstellung mit RBAC aktivierter Option CloudWatch Metriken, um Benutzer über Sicherheitsereignisse zu informieren. Diese Metriken helfen dabei, fehlgeschlagene Versuche zu identifizieren, auf Schlüssel zuzugreifen oder Befehle auszuführen, für die RBAC Benutzer, für die eine Verbindung hergestellt wird, nicht autorisiert sind.

Darüber hinaus tragen AWS Produkt- und Servicere Ressourcen dazu bei, Ihre gesamte Arbeitslast zu sichern, indem sie Bereitstellungen automatisieren und alle Aktionen und Änderungen für eine spätere Überprüfung/Prüfung protokollieren.

Vorteil auf Fragenebene: Durch die Überwachung von Ereignissen ermöglichen Sie Ihrem Unternehmen, gemäß Ihren Anforderungen, Richtlinien und Verfahren zu reagieren. Durch die Automatisierung der Überwachung und Reaktion auf diese Sicherheitsereignisse wird Ihre allgemeine Sicherheitslage gestärkt.

- [Erforderlich] Machen Sie sich mit den veröffentlichten CloudWatch Kennzahlen zu RBAC Authentifizierungs- und Autorisierungsfehlern vertraut.
  - AuthenticationFailures = Fehlgeschlagene Versuche, sich bei Valkey oder Redis zu authentifizieren OSS

- `KeyAuthorizationFailures` = Fehlgeschlagene Versuche von Benutzern, ohne Erlaubnis auf Schlüssel zuzugreifen
- `CommandAuthorizationFailures` = Fehlgeschlagene Versuche von Benutzern, Befehle ohne Erlaubnis auszuführen

[Ressourcen]:

- [Metriken für Valkey oder Redis OSS](#)
- [Am besten] Es wird empfohlen, Warnmeldungen und Benachrichtigungen für diese Metriken einzurichten und je nach Bedarf zu reagieren.

[Ressourcen]:

- [CloudWatch Amazon-Alarme verwenden](#)
- [Am besten] Verwenden Sie den OSS `ACL LOG` Befehl Valkey oder Redis, um weitere Informationen zu sammeln

[Ressourcen]:

- [ACL LOG](#)
- [Am besten] Machen Sie sich mit den Funktionen der AWS Produkte und Services im Zusammenhang mit der Überwachung, Protokollierung und Analyse von ElastiCache Bereitstellungen und Ereignissen vertraut

[Ressourcen]:

- [ElastiCache API Amazon-Anrufe protokollieren mit AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-Backup-Check](#)
- [Überwachung der Nutzung mit Metrics CloudWatch](#)

## Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens

Der Schwerpunkt der Zuverlässigkeit liegt auf Workloads, die ihre vorgesehenen Funktionen erfüllen, und darauf, wie sie bei einem Ausfall schnell wiederhergestellt werden können, um die Anforderungen zu erfüllen. Zu den wichtigsten Themen gehören der Entwurf verteilter Systeme, die Wiederherstellungsplanung und die Anpassung an sich ändernde Anforderungen.

Themen

- [REL1: Wie unterstützen Sie Implementierungen von Hochverfügbarkeitsarchitekturen \(HA\)?](#)

- [REL2: Wie erreichen Sie Ihre Recovery Point Objectives \(RPOs\)? ElastiCache](#)
- [REL3: Wie unterstützen Sie die Anforderungen für die Notfallwiederherstellung \(DR\)?](#)
- [REL4: Wie können Sie Failovers effektiv planen?](#)
- [REL5: Sind Ihre ElastiCache Komponenten so konzipiert, dass sie skalierbar sind?](#)

## REL1: Wie unterstützen Sie Implementierungen von Hochverfügbarkeitsarchitekturen (HA)?

Einführung auf Fragenebene: Wenn Sie die Hochverfügbarkeitsarchitektur von Amazon verstehen, können ElastiCache Sie bei Verfügbarkeitsereignissen in einem stabilen Zustand arbeiten.

Vorteil auf Fragenebene: Wenn Sie Ihre ElastiCache Cluster so gestalten, dass sie ausfallsicher sind, wird eine höhere Verfügbarkeit Ihrer Bereitstellungen gewährleistet. ElastiCache

- [Erforderlich] Ermitteln Sie das Maß an Zuverlässigkeit, das Sie für Ihren Cluster benötigen. ElastiCache Verschiedene Workloads haben unterschiedliche Resilienzstandards, von rein kurzlebigen bis hin zu geschäftskritischen Workloads. Definieren Sie die Anforderungen für jede Art von Umgebung, die Sie betreiben, z. B. Entwicklung, Test und Produktion.

Caching-Engine: ElastiCache (Memcached) gegen ElastiCache (Redis) OSS

1. ElastiCache (Memcached) bietet keinen Replikationsmechanismus und wird hauptsächlich für kurzlebige Workloads verwendet.
  2. ElastiCache (RedisOSS) bietet HA-Funktionen, die im Folgenden beschrieben werden
- [Am besten] Verwenden Sie ElastiCache (RedisOSS) für Workloads, die HA erfordern, im Clustermodus mit mindestens zwei Replikaten pro Shard, auch für Workloads mit geringem Durchsatz, die nur einen Shard benötigen.
    1. Wenn der Clustermodus aktiviert ist, wird Multi-AZ automatisch aktiviert.

Multi-AZ minimiert Ausfallzeiten, indem automatische Failovers vom Primärknoten bis zu den Replikaten im Falle einer geplanten oder ungeplanten Wartung durchgeführt und AZ-Ausfälle minimiert werden.

2. Bei Shard-Workloads sorgen mindestens drei Shards für eine schnellere Wiederherstellung bei Failover-Ereignissen, da das Valkey- oder OSS Redis-Cluster-Protokoll erfordert, dass die Mehrheit der Primärknoten verfügbar ist, um ein Quorum zu erreichen.
3. Richten Sie zwei oder mehr Replikate für die Verfügbarkeit ein.

Zwei Replikate bieten eine verbesserte Leseskalierbarkeit sowie Leseverfügbarkeit in Szenarien, in denen ein Replikat gewartet wird.

4. Verwenden Sie Graviton2-basierte Knotentypen (Standardknoten in den meisten Regionen).

ElastiCache (RedisOSS) hat die Leistung auf diesen Knoten optimiert. Dadurch erhalten Sie eine bessere Replikations- und Synchronisierungsleistung, was zu einer insgesamt verbesserten Verfügbarkeit führt.

5. Überwachen Sie und passen Sie die Größe an, um auf erwartete Verkehrsspitzen zu reagieren: Bei hoher Auslastung reagiert die ElastiCache (Redis-OSS) Engine möglicherweise nicht mehr, was sich auf die Verfügbarkeit auswirkt. `BytesUsedForCache` und `DatabaseMemoryUsagePercentage` sind gute Indikatoren für Ihre Speichernutzung, wohingegen `ReplicationLag` sie anhand Ihrer Schreibrate ein Indikator für den Zustand Ihrer Replikation sind. Sie können diese Metriken verwenden, um die Clusterskalierung auszulösen.

6. Stellen Sie die Stabilität auf der Clientseite sicher, indem Sie [APIvor einem Produktions-Failover-Ereignis Tests mit dem Failover](#) durchführen.

[Ressourcen]:

- [Konfigurieren Sie ElastiCache \(Redis\) für eine höhere Verfügbarkeit OSS](#)
- [Hohe Verfügbarkeit mit Replikationsgruppen](#)

## REL2: Wie erreichen Sie Ihre Recovery Point Objectives (RPOs)? ElastiCache

Einführung auf Fragenebene: Machen Sie sich mit der Arbeitslast vertraut RPO, um fundierte Entscheidungen über ElastiCache Sicherheits- und Wiederherstellungsstrategien treffen zu können.

Vorteil auf Fragenebene: Mit einer RPO Strategie vor Ort kann die Geschäftskontinuität im Fall von Notfallwiederherstellungsszenarien verbessert werden. Die Gestaltung Ihrer Sicherheits- und Wiederherstellungsrichtlinien kann Ihnen helfen, Ihre Recovery Point Objectives (RPO) für Ihre Daten zu erreichen. ElastiCache ElastiCache (RedisOSS) bietet Snapshot-Funktionen, die in Amazon S3 gespeichert werden, zusammen mit einer konfigurierbaren Aufbewahrungsrichtlinie. Diese Snapshots werden während eines definierten Backup-Fensters aufgenommen und vom Service automatisch verarbeitet. Wenn Ihr Workload zusätzliche Backup-Granularität erfordert, haben Sie die Möglichkeit, bis zu 20 manuelle Backups pro Tag zu erstellen. Manuell erstellte Backups unterliegen keiner Serviceaufbewahrungsrichtlinie und können auf unbestimmte Zeit aufbewahrt werden.

- [Erforderlich] Machen Sie sich mit Ihren ElastiCache Bereitstellungen vertraut und dokumentieren Sie sie. RPO
  - Beachten Sie, dass Memcached keine Backup-Prozesse anbietet.
  - Informieren Sie sich über die Funktionen der ElastiCache Backup- und Wiederherstellungsfunktionen.
- [Am besten] Etablieren Sie einen gut kommunizierten Prozess für die Sicherung Ihres Clusters.
  - Initiieren Sie bei Bedarf manuelle Backups.
  - Überprüfen Sie die Aufbewahrungsrichtlinien für automatische Backups.
  - Beachten Sie, dass manuelle Backups auf unbestimmte Zeit aufbewahrt werden.
  - Planen Sie Ihre automatischen Backups in Zeiten geringer Auslastung.
  - Führen Sie Backup-Operationen anhand von Lesereplikaten durch, um sicherzustellen, dass die Auswirkungen auf die Cluster-Leistung so gering wie möglich gehalten werden.
- [Gut] Nutzen Sie die Funktion für geplante Backups von ElastiCache , um Ihre Daten regelmäßig innerhalb eines bestimmten Zeitfensters zu sichern.
  - Testen Sie regelmäßig Wiederherstellungen aus Ihren Backups.
- [Ressourcen]:
  - [Redis OSS](#)
  - [Backup und Wiederherstellung für ElastiCache \(RedisOSS\)](#)
  - [Erstellen manueller Backups](#)
  - [Planen automatischer Backups](#)
  - [Cluster Backup und wiederherstellen ElastiCache \(RedisOSS\)](#)

### REL3: Wie unterstützen Sie die Anforderungen für die Notfallwiederherstellung (DR)?

Einführung auf Fragenebene: Die Notfallwiederherstellung ist ein wichtiger Aspekt jeder Workload-Planung. ElastiCache (RedisOSS) bietet mehrere Optionen zur Implementierung von Disaster Recovery auf der Grundlage von Anforderungen an die Belastbarkeit von Workloads. Mit Amazon ElastiCache Global Datastore können Sie in Ihren ElastiCache (Redis-OSS) Cluster in einer Region schreiben und die Daten zum Lesen aus zwei anderen regionsübergreifenden Replikatclustern zur Verfügung stellen, wodurch Lesevorgänge mit niedriger Latenz und regionsübergreifende Disaster Recovery ermöglicht werden.

Vorteil auf Fragenebene: Durch das Nachvollziehen und die Planung verschiedener Notfallszenarien kann die Geschäftskontinuität gewährleistet werden. DR-Strategien müssen gegenüber Kosten, Leistungseinbußen und Datenverlustpotenzial abgewogen werden.

- [Erforderlich] Entwickeln und dokumentieren Sie DR-Strategien für all Ihre ElastiCache Komponenten auf der Grundlage der Workload-Anforderungen. ElastiCache ist insofern einzigartig, als einige Anwendungsfälle völlig kurzlebig sind und keine DR-Strategie erfordern, während andere am anderen Ende des Spektrums liegen und eine extrem robuste DR-Strategie erfordern. Alle Optionen müssen gegenüber Kostenoptimierung abgewogen werden – eine größere Resilienz erfordert eine umfassendere Infrastruktur.

Machen Sie sich mit den DR-Optionen vertraut, die auf regionaler und multiregionaler Ebene verfügbar sind.

- Multi-AZ-Bereitstellungen werden empfohlen, um sich vor AZ-Ausfällen zu schützen. Stellen Sie sicher, dass bei der Bereitstellung der Cluster-Modus in Multi-AZ-Architekturen aktiviert ist und mindestens 3 verfügbar sind. AZs
- Global Datastore wird als Schutz vor regionalen Ausfällen empfohlen.
- [Am besten] Aktivieren Sie Global Datastore für Workloads, die Resilienz auf regionaler Ebene erfordern.
  - Planen Sie ein Failover auf die sekundäre Region für den Fall einer Verschlechterung der primären Region ein.
  - Testen Sie den Failover-Prozess für mehrere Regionen, bevor Sie ein Failover in der Produktion durchführen.
  - Überwachen Sie die ReplicationLag-Metrik, um die möglichen Auswirkungen von Datenverlusten bei Failover-Ereignissen zu verstehen.
- [Ressourcen]:
  - [Minimieren von Ausfällen](#)
  - [Regionsübergreifende Replikation mithilfe globaler Datenspeicher AWS](#)
  - [Wiederherstellen aus einem Backup mit optionaler Cluster-Größenanpassung](#)
  - [Minimierung der Ausfallzeiten in ElastiCache \(OSSRedis\) mit Multi-AZ](#)

## REL4: Wie können Sie Failovers effektiv planen?

Einführung auf Fragenebene: Die Aktivierung von Multi-AZ mit automatischen Failovers ist eine bewährte Methode. ElastiCache In bestimmten Fällen ersetzt ElastiCache (RedisOSS) die

Primärknoten im Rahmen von Servicebetrieben. Beispiele hierfür sind geplante Wartungsereignisse und der unwahrscheinliche Falle eines Ausfalls des Knotens oder eines Problems mit der Availability Zone. Erfolgreiche Failover hängen sowohl ElastiCache von beiden als auch von Ihrer Client-Bibliothekskonfiguration ab.

Vorteil auf Fragenebene: Wenn Sie die Best Practices für ElastiCache Failover in Verbindung mit Ihrer spezifischen ElastiCache (Redis-OSS) Clientbibliothek befolgen, können Sie potenzielle Ausfallzeiten bei Failover-Ereignissen minimieren.

- [Erforderlich] Verwenden Sie bei deaktiviertem Clustermodus Timeouts, damit Ihre Clients anhand der aktualisierten IP-Adresse des primären Endpunkts erkennen, ob die Verbindung zum alten Primärknoten getrennt und erneut eine Verbindung zum neuen Primärknoten hergestellt werden muss. Im aktivierten Clustermodus ist die Clientbibliothek dafür zuständig, Änderungen in der zugrundeliegenden Cluster-Topologie zu erkennen. Dies wird meistens durch Konfigurationseinstellungen in der ElastiCache (Redis-OSS) Clientbibliothek erreicht, mit denen Sie auch die Häufigkeit und die Methode der Aktualisierung konfigurieren können. Jede Clientbibliothek bietet eigene Einstellungen. Weitere Informationen finden Sie in der entsprechenden Dokumentation.

[Ressourcen]:

- [Minimierung der Ausfallzeiten in ElastiCache \(RedisOSS\) mit Multi-AZ](#)
- Informieren Sie sich über die Best Practices Ihrer ElastiCache (Redis OSS -) Client-Bibliothek.
- [Erforderlich] Erfolgreiche Failover hängen von einer fehlerfreien Replikationsumgebung zwischen dem Primärknoten und den Replikatknoten ab. Informieren Sie sich über die asynchrone Natur der Valkey- und OSS Redis-Replikation sowie über die verfügbaren CloudWatch Metriken zur Berichterstattung über die Replikationsverzögerung zwischen Primär- und Replikatknoten. In Anwendungsfällen, die eine höhere Datensicherheit erfordern, können Sie den WAIT Befehl nutzen, um Replikate dazu zu zwingen, Schreibvorgänge zu bestätigen, bevor sie auf verbundene Clients antworten.

[Ressourcen]:

- [Metriken für Valkey oder Redis OSS](#)
- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) unter Verwendung von Amazon CloudWatch](#)
- [Am besten] Überprüfen Sie regelmäßig die Reaktionsfähigkeit Ihrer Anwendung während eines Failovers mithilfe des ElastiCache Test-Failovers. API

[Ressourcen]:

- [Testen des automatischen Failovers auf eine Read Replica auf Amazon ElastiCache \(Redis\) OSS](#)
- [Testen des automatischen Failovers](#)

REL5: Sind Ihre ElastiCache Komponenten so konzipiert, dass sie skalierbar sind?

Einführung auf Fragenebene: Wenn Sie die Skalierungsmöglichkeiten und die verfügbaren Bereitstellungstopologien verstehen, können sich Ihre ElastiCache Komponenten im Laufe der Zeit an sich ändernde Workload-Anforderungen anpassen. ElastiCache bietet 4-Wege-Skalierung: ein-/auswärts (horizontal) sowie hoch/runter (vertikal).

Vorteil auf Frageebene: Die Einhaltung von Best Practices für ElastiCache Bereitstellungen bietet ein Höchstmaß an Flexibilität bei der Skalierung und erfüllt gleichzeitig das Well-Architected-Prinzip der horizontalen Skalierung, um die Auswirkungen von Ausfällen zu minimieren.

- [Erforderlich] Machen Sie sich mit dem Unterschied zwischen Topologien im aktivierten und im deaktivierten Clustermodus vertraut. In fast allen Fällen wird empfohlen, die Bereitstellung mit aktiviertem Clustermodus vorzunehmen, da dies im Zeitverlauf eine größere Skalierbarkeit ermöglicht. Komponenten im deaktiviertem Clustermodus können nur begrenzt horizontal skaliert werden, indem Lesereplikate hinzugefügt werden.
- [Erforderlich] Machen Sie sich damit vertraut, wann und wie Sie skalieren müssen.
  - Weitere Informationen: Fügen Sie Replikate hinzu READIOPS
  - Weitere Informationen WRITEOPS: Fügen Sie Shards hinzu (horizontal skalieren)
  - Für mehr Netzwerk-I/O – netzwerkoptimierte Instances verwenden, hochskalieren
- [Am besten] Stellen Sie Ihre ElastiCache Komponenten mit aktiviertem Cluster-Modus bereit, wobei der Schwerpunkt eher auf mehr, kleinere Knoten als auf weniger, größere Knoten liegt. Dadurch wird der Explosionsradius eines Knotenausfalls effektiv begrenzt.
- [Am besten] Nehmen Sie Replikate in Ihre Cluster auf, um die Reaktionsfähigkeit bei Skalierungsereignissen zu verbessern
- [Gut] Verwenden Sie bei deaktiviertem Cluster-Modus Read Replicas, um die gesamte Lesekapazität zu erhöhen. ElastiCache unterstützt bis zu 5 Read Replicas im deaktivierten Cluster-Modus sowie vertikale Skalierung.
- [Ressourcen]:

- [Skalierung von Clustern ElastiCache \(Redis\) OSS](#)
- [Online-Hochskalieren](#)
- [Skalierung ElastiCache für Memcached-Cluster](#)

## Säule der Leistung und Effizienz von Amazon ElastiCache Well-Architected Lens

Die Säule der Leistungseffizienz konzentriert sich auf die effiziente Nutzung von IT- und Computing-Ressourcen. Zu den wichtigsten Themen gehören die Auswahl der richtigen Ressourcentypen und -größen je nach Workload-Anforderungen, die Überwachung der Leistung und das Treffen fundierter Entscheidungen, um die Effizienz aufrechtzuerhalten, wenn sich die Geschäftsanforderungen weiterentwickeln.

### Themen

- [PE 1: Wie überwachen Sie die Leistung Ihres ElastiCache Amazon-Clusters?](#)
- [PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache Clusterknoten?](#)
- [PE 3: Wie verfolgen und protokollieren Sie bei Caching-Workloads die Effektivität und Leistung Ihres Caches?](#)
- [PE 4: Wie optimiert Ihr Workload den Einsatz von Netzwerkressourcen und Verbindungen?](#)
- [PE 5: Wie verwalten Sie die Löschung und/oder Bereinigung von Schlüsseln?](#)
- [PE 6: Wie modellieren und interagieren Sie mit Daten in ElastiCache?](#)
- [PE 7: Wie protokollieren Sie langsam laufende Befehle in Ihrem ElastiCache Amazon-Cluster?](#)
- [PE8: Wie hilft Auto Scaling dabei, die Leistung des ElastiCache Clusters zu steigern?](#)

### PE 1: Wie überwachen Sie die Leistung Ihres ElastiCache Amazon-Clusters?

Einführung auf Fragenebene: Wenn Sie die vorhandenen Überwachungsmetriken verstehen, können Sie die aktuelle Auslastung ermitteln. Eine ordnungsgemäße Überwachung kann dazu beitragen, potenzielle Engpässe zu identifizieren, die sich auf die Leistung Ihres Clusters auswirken.

Vorteil auf Fragenebene: Wenn Sie die Metriken Ihres Clusters kennen, können Sie Optimierungsmethoden entwickeln, die zu einer geringeren Latenz und einem höheren Durchsatz führen können.

- [Erforderlich] Basisleistungstests mit einer Teilmenge Ihres Workloads.

- Sie sollten die Leistung des tatsächlichen Workloads mithilfe von Mechanismen wie Lasttests überwachen.
- Überwachen Sie die CloudWatch Metriken, während Sie diese Tests durchführen, um sich ein Bild von den verfügbaren Metriken zu machen und eine Leistungsbasis festzulegen.
- [Am besten] Benennen Sie für ElastiCache (Redis-OSS) Workloads rechenintensive Befehle um, z. B. um die Fähigkeit von Benutzern einzuschränken KEYS, blockierende Befehle auf Produktionsclustern auszuführen.
- ElastiCache (Redis-OSS) Workloads, auf denen Engine 6.x ausgeführt wird, können die rollenbasierte Zugriffskontrolle nutzen, um bestimmte Befehle einzuschränken. Der Zugriff auf die Befehle kann gesteuert werden, indem Benutzer und Benutzergruppen mit der AWS Konsole erstellt oder CLI die Benutzergruppen einem (Redis-) Cluster zugeordnet werden. ElastiCache OSS In Redis OSS 6 können wir, wenn es aktiviert RBAC ist, „- @dangerous“ verwenden, wodurch teure Befehle wie KEYS, MONITOR SORT, usw. für diesen Benutzer nicht zugelassen werden.
- Benennen Sie Befehle für Engine-Version 5.x mithilfe des `rename-command` Parameters in der Cluster-Parametergruppe ElastiCache (RedisOSS) um.
- [Besser] Analysieren Sie langsame Abfragen und suchen Sie nach Optimierungsmöglichkeiten.
  - Bei ElastiCache (Redis-OSS) Workloads erfahren Sie mehr über Ihre Abfragen, indem Sie das Slow Log analysieren. Sie können beispielsweise den Befehl `valkey-cli slowlog get 10` verwenden, um die letzten 10 Befehle anzuzeigen, welche die Latenzschwellenwerte überschritten haben (standardmäßig 10 Sekunden).
  - Bestimmte Abfragen können mithilfe komplexer ElastiCache (Redis-OSS) Datenstrukturen effizienter ausgeführt werden. Für Bereichsabfragen im numerischen Stil kann eine Anwendung beispielsweise einfache numerische Indizes mit sortierten Sätzen implementieren. Die Verwaltung dieser Indizes kann die Anzahl der am Datensatz durchgeführten Scans reduzieren und Daten mit höherer Leistungseffizienz zurückgeben.
  - `redis-benchmark` bietet für ElastiCache (Redis-OSS) Workloads eine einfache Oberfläche zum Testen der Leistung verschiedener Befehle mithilfe benutzerdefinierter Eingaben wie Anzahl der Clients und Datengröße.
  - Da Memcached nur einfache Befehle auf Schlüsselebene unterstützt, sollten Sie erwägen, zusätzliche Schlüssel als Indizes zu erstellen, um zu vermeiden, dass der Schlüsselbereich wiederholt wird, um Client-Abfragen zu bearbeiten.
- [Ressourcen]:
  - [Überwachung der Nutzung mit Metrics CloudWatch](#)

- [CloudWatch Amazon-Alarme verwenden](#)
- [Valkey- und Redis-spezifische Parameter OSS](#)
- [SLOWLOG](#)
- [Benchmark](#)

## PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache Clusterknoten?

Einführung auf Fragenebene: Die Art und Weise, wie Ihre Anwendung eine Verbindung zu ElastiCache Amazon-Knoten herstellt, kann sich auf die Leistung und Skalierbarkeit des Clusters auswirken.

Vorteil auf Fragenebene: Durch die richtige Nutzung der verfügbaren Knoten im Cluster wird sichergestellt, dass die Arbeit auf die verfügbaren Ressourcen verteilt wird. Die folgenden Methoden tragen ebenfalls dazu bei, ungenutzte Ressourcen zu vermeiden.

- [Erforderlich] Sorgen Sie dafür, dass sich die Clients mit dem richtigen ElastiCache Endpunkt verbinden.
  - ElastiCache (RedisOSS) implementiert je nach verwendetem Clustermodus unterschiedliche Endpunkte. Wenn der Clustermodus aktiviert ist, ElastiCache wird ein Konfigurationsendpunkt bereitgestellt. ElastiCache stellt bei deaktiviertem Clustermodus einen primären Endpunkt bereit, der normalerweise für Schreibvorgänge verwendet wird, und einen Leser-Endpunkt für die Verteilung von Lesevorgängen zwischen Replikaten. Die korrekte Implementierung dieser Endpunkte führt zu einer besseren Leistung und einfacheren Skalierungsvorgängen. Vermeiden Sie es, eine Verbindung zu einzelnen Knotenendpunkten herzustellen, es sei denn, dies ist ausdrücklich erforderlich.
  - ElastiCache stellt für Memcached-Cluster mit mehreren Knoten einen Konfigurationsendpunkt bereit, der Auto Discovery aktiviert. Es wird empfohlen, einen Hashing-Algorithmus zu verwenden, um die Arbeit gleichmäßig auf die Cache-Knoten zu verteilen. Viele Memcached-Clientbibliotheken implementieren konsistentes Hashing. Überprüfen Sie die Dokumentation der von Ihnen verwendeten Bibliothek darauf, ob sie konsistentes Hashing unterstützt und wie es implementiert wird. Weitere Informationen zur Implementierung dieser Funktionen finden Sie [hier](#).
- [Besser] Nutzen Sie den aktivierten Clustermodus ElastiCache (RedisOSS), um die Skalierbarkeit zu verbessern.
  - ElastiCache (RedisOSS) -Cluster (Clustermodus aktiviert) unterstützen [Online-Skalierungsoperationen](#) (ausgehen/einsteigen und hoch/runter), um Daten dynamisch auf

mehrere Shards zu verteilen. Durch die Verwendung des Konfigurationsendpunkts wird sichergestellt, dass sich Ihre clusterfähigen Clients an Änderungen in der Cluster-Topologie anpassen können.

- Sie können den Cluster auch neu verteilen, indem Sie Hashslots zwischen verfügbaren Shards in Ihrem Cluster ElastiCache (OSSRedis) (Clustermodus aktiviert) verschieben. Dies trägt dazu bei, die Arbeit effizienter auf die verfügbaren Shards zu verteilen.
- [Besser] Implementieren Sie eine Strategie zur Identifizierung und Korrektur von Tastaturbefehlen in Ihrem Workload.
  - Bedenken Sie die Auswirkungen multidimensionaler Valkey- oder OSS Redis-Datenstrukturen wie Listen, Streams, Sets usw. Diese Datenstrukturen werden in einzelnen Schlüsseln gespeichert, die sich auf einem einzigen Knoten befinden. Ein sehr großer multidimensionaler Schlüssel nimmt potenziell mehr Netzwerkkapazität und Speicher in Anspruch als andere Datentypen und kann eine unverhältnismäßige Nutzung dieses Knotens bewirken. Wenn möglich, sollten Sie Ihren Workload so gestalten, dass der Datenzugriff auf viele einzelne Schlüssel verteilt wird.
  - Tastaturbefehle im Workload können die Leistung des verwendeten Knotens beeinträchtigen. Bei ElastiCache (Redis-OSS) Workloads können Sie Hotkeys erkennen, `valkey-cli --hotkeys` wenn eine Richtlinie zur maximalen LFU Speicherbelegung vorhanden ist.
  - Erwägen Sie, Tastaturbefehle auf mehreren Knoten zu replizieren, um den Zugriff auf sie gleichmäßiger zu verteilen. Bei diesem Ansatz muss der Client auf mehrere Primärknoten schreiben (der Valkey- oder OSS Redis-Knoten selbst bietet diese Funktionalität nicht) und zusätzlich zum ursprünglichen Schlüsselnamen eine Liste mit Schlüsselnamen zum Lesen führen.
  - ElastiCache [mit Valkey 7.2 und höher und Redis OSS Version 6 und höher unterstützen servergestütztes clientseitiges Caching](#). Auf diese Weise können Anwendungen auf Änderungen an einem Schlüssel warten, bevor sie über das Netzwerk zurückrufen. ElastiCache
- [Ressourcen]:
  - [Konfigurieren Sie ElastiCache mit Valkey und Redis OSS für eine höhere Verfügbarkeit](#)
  - [Verbindungsendpunkte finden in ElastiCache](#)
  - [Bewährte Methoden für den Lastausgleich](#)
  - [Online-Resharding für Valkey oder OSS Redis \(Clustermodus aktiviert\)](#)
  - [Clientseitiges Caching in Valkey und Redis OSS](#)

## PE 3: Wie verfolgen und protokollieren Sie bei Caching-Workloads die Effektivität und Leistung Ihres Caches?

Einführung auf Fragenebene: Caching ist ein häufig auftretender Workload, ElastiCache und es ist wichtig, dass Sie wissen, wie Sie die Effektivität und Leistung Ihres Caches verwalten können.

Vorteil auf Fragenebene: Ihre Anwendung kann Anzeichen einer stagnierenden Leistung aufweisen. Ihre Fähigkeit, Cache-spezifische Metriken als Grundlage für Ihre Entscheidung zur Steigerung der App-Leistung zu verwenden, ist für Ihren Cache-Workload von entscheidender Bedeutung.

- [Erforderlich] Messen und verfolgen Sie die Cache-Trefferquote im Zeitverlauf. Die Effizienz Ihres Caches wird durch seine „Cache-Trefferquote“ bestimmt. Die Cache-Trefferquote wird durch die Summe der Schlüsseltreffer geteilt durch die Gesamtzahl der Treffer und Fehlversuche definiert. Je näher die Quote an 1 liegt, desto effektiver ist Ihr Cache. Eine niedrige Cache-Trefferquote wird durch die Menge der Cache-Fehlversuche verursacht. Cache-Fehlversuche treten auf, wenn der angeforderte Schlüssel nicht im Cache gefunden wird. Ein Schlüssel befindet sich nicht im Cache, weil er entweder bereinigt oder gelöscht wurde, abgelaufen ist oder nie existiert hat. Machen Sie sich klar, warum sich Schlüssel nicht im Cache befinden, und entwickeln Sie geeignete Strategien, um sie im Cache vorliegen zu haben.

[Ressourcen]:

- [Metriken für Valkey und Redis OSS](#)
- [Erforderlich] Messen und erfassen Sie die Leistung Ihres Anwendungs-Caches in Verbindung mit Latenz- und CPU Nutzungswerten, um zu ermitteln, ob Sie Anpassungen an Ihren time-to-live oder anderen Anwendungskomponenten vornehmen müssen. ElastiCache bietet eine Reihe von CloudWatch Metriken für aggregierte Latenzen für jede Datenstruktur. Diese Latenzmetriken werden anhand der Commandstats-Statistik des INFO Befehls ElastiCache (RedisOSS) berechnet und beinhalten nicht die Netzwerk- und I/O-Zeit. Dies ist nur die Zeit, die ElastiCache (RedisOSS) für die Verarbeitung der Operationen benötigt.

[Ressourcen]:

- [Metriken für Valkey und Redis OSS](#)
- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) unter Verwendung von Amazon CloudWatch](#)
- [Am besten] Wählen Sie die richtige Caching-Strategie für Ihre Anforderungen aus. Eine niedrige Cache-Trefferquote wird durch die Menge der Cache-Fehlversuche verursacht. Wenn Ihr Workload so konzipiert ist, dass es nur wenige Cache-Fehlversuche gibt (z. B. Kommunikation

in Echtzeit), sollten Sie Ihre Caching-Strategien überprüfen und die für Ihren Workload am besten geeigneten Lösungen anwenden, z. B. Abfrage-Instrumente zur Messung von Speicher und Leistung. Die tatsächlichen Strategien, die Sie zum Auffüllen und Verwalten Ihres Cache implementieren müssen, hängt von den Daten, die Ihre Clients zwischenspeichern müssen, und den Zugriffsmustern auf diese Daten ab. Es ist beispielsweise unwahrscheinlich, dass Sie dieselbe Strategie sowohl für personalisierte Empfehlungen in einer Streaming-Anwendung als auch für aktuelle Nachrichten verwenden.

[Ressourcen]:

- [Caching-Strategien für Memcached](#)
- [Bewährte Methoden für das Caching](#)
- [Skalierbare Leistung mit Amazon ElastiCache Whitepaper](#)

## PE 4: Wie optimiert Ihr Workload den Einsatz von Netzwerkressourcen und Verbindungen?

Einführung auf Fragenebene: ElastiCache (RedisOSS) und ElastiCache (Memcached) werden von vielen Anwendungsclients unterstützt, und die Implementierungen können variieren. Sie müssen das vorhandene Netzwerk- und Verbindungsmanagement verstehen, um mögliche Auswirkungen auf die Leistung zu analysieren.

Vorteil auf Fragenebene: Die effiziente Nutzung von Netzwerkressourcen kann die Leistungseffizienz Ihres Clusters verbessern. Die folgenden Empfehlungen können die Netzwerkanforderungen reduzieren und die Latenz und den Durchsatz des Clusters verbessern.

- [Erforderlich] Verwalten Sie proaktiv Verbindungen zu Ihrem Cluster. ElastiCache
  - Verbindungspooling in der Anwendung reduziert den Overhead im Cluster, der durch das Öffnen und Schließen von Verbindungen entsteht. Überwachen Sie das Verbindungsverhalten in Amazon CloudWatch mithilfe von `CurConnections` und `NewConnections`.
  - Vermeiden Sie Verbindungslecks, indem Sie die Client-Verbindungen gegebenenfalls ordnungsgemäß schließen. Zu den Strategien zur Verbindungsverwaltung gehören das ordnungsgemäße Schließen von Verbindungen, die nicht verwendet werden, und das Festlegen von Verbindungs-Timeouts.
  - Für Memcached-Workloads gibt es eine konfigurierbare Speichermenge, die für die Verarbeitung von Verbindungen reserviert ist und als `memcached_connections_overhead` bezeichnet wird.

- [Besser] Komprimieren Sie große Objekte, um den Speicherplatz zu reduzieren und den Netzwerkdurchsatz zu verbessern.
  - Datenkomprimierung kann den erforderlichen Netzwerkdurchsatz (Gbit/s) reduzieren, erhöht jedoch den Arbeitsaufwand für die Anwendung zum Komprimieren und Dekomprimieren von Daten.
  - Durch die Komprimierung wird auch der Speicherverbrauch von Schlüsseln reduziert.
  - Berücksichtigen Sie auf der Grundlage Ihrer Anwendungsanforderungen die Kompromisse zwischen Komprimierungsverhältnis und Komprimierungsgeschwindigkeit.
- [Ressourcen]:
  - [ElastiCache \(RedisOSS\) — Globaler Datenspeicher](#)
  - [Memcached-spezifische Parameter](#)
  - [ElastiCache \(RedisOSS\) 5.0.3 verbessert die I/O-Verarbeitung, um die Leistung zu steigern](#)
  - [Metriken für Valkey und Redis OSS](#)
  - [Konfigurieren Sie ElastiCache \(RedisOSS\) für eine höhere Verfügbarkeit](#)

## PE 5: Wie verwalten Sie die Löschung und/oder Bereinigung von Schlüsseln?

Einführung auf Fragenebene: Workloads haben unterschiedliche Anforderungen und erwartetes Verhalten, wenn sich ein Clusterknoten den Grenzwerten für den Speicherverbrauch nähert. ElastiCache (RedisOSS) hat unterschiedliche Richtlinien für den Umgang mit diesen Situationen.

Vorteil auf Fragenebene: Die richtige Verwaltung des verfügbaren Speichers und das Verständnis der Lösungsrichtlinien tragen dazu bei, dass das Clusterverhalten bei Überschreitung der Speicherlimits von Instances besser erkannt wird.

- [Erforderlich] Instrumentieren Sie den Datenzugriff, um zu beurteilen, welche Richtlinie angewendet werden soll. Identifizieren Sie eine geeignete Richtlinie für maximalen Arbeitsspeicher, um zu kontrollieren, ob und wie Bereinigungen im Cluster durchgeführt werden.
  - Eine Bereinigung erfolgt, wenn der maximale Arbeitsspeicher des Clusters verbraucht ist und eine Richtlinie vorhanden ist, welche die Bereinigung zulässt. Das Verhalten des Clusters in dieser Situation hängt von der angegebenen Bereinigungsrichtlinie ab. Diese Richtlinie kann mithilfe der Parametergruppe `maxmemory-policy` On the ElastiCache (RedisOSS) -Cluster verwaltet werden.

- Die Standardrichtlinie gibt Speicherplatz `volatile-lru` frei, indem Schlüssel mit einer festgelegten Ablaufzeit (Wert) entfernt werden. TTL In den Richtlinien „Am wenigsten häufig verwendet“ (LFU) und „Zuletzt verwendet“ (LRU) werden Schlüssel je nach Nutzung entfernt.
- Für Memcached-Workloads gibt es eine LRU Standardrichtlinie, die Räumungen auf jedem Knoten steuert. Die Anzahl der Räumungen in Ihrem ElastiCache Amazon-Cluster kann mithilfe der Räumungsmetrik auf Amazon überwacht werden. CloudWatch
- [Besser] Standardisieren Sie das Löschverhalten, um die Auswirkungen auf die Leistung Ihres Clusters zu kontrollieren und unerwartete Leistungsengpässe zu vermeiden.
  - Bei ElastiCache (Redis-OSS) Workloads ist das explizite Entfernen von Schlüsseln aus dem Cluster wie folgt DEL: UNLINK Es entfernt die angegebenen Schlüssel. Der Befehl führt jedoch die eigentliche Speicherrückgewinnung in einem anderen Thread durch, sodass er nicht blockiert, während dies mit DEL der Fall ist. Die eigentliche Entfernung erfolgt später asynchron.
  - Für ElastiCache (RedisOSS) 6.x-Workloads kann das Verhalten des DEL Befehls in der Parametergruppe mithilfe von Parameter geändert werden. `lazyfree-lazy-user-del`
- [Ressourcen]:
  - [Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen](#)
  - [UNLINK](#)
  - [Cloud-Finanzmanagement mit AWS](#)

## PE 6: Wie modellieren und interagieren Sie mit Daten in ElastiCache?

Einführung auf Fragenebene: ElastiCache ist stark von der Anwendung abhängig von den Datenstrukturen und dem verwendeten Datenmodell, muss aber auch der zugrunde liegende Datenspeicher (falls vorhanden) berücksichtigt werden. Machen Sie sich mit den verfügbaren ElastiCache (Redis-OSS) Datenstrukturen vertraut und stellen Sie sicher, dass Sie die für Ihre Bedürfnisse am besten geeigneten Datenstrukturen verwenden.

Vorteil auf Fragenebene: Die Datenmodellierung ElastiCache besteht aus mehreren Ebenen, einschließlich Anwendungsfällen, Datentypen und Beziehungen zwischen Datenelementen. Darüber hinaus haben alle ElastiCache (Redis-OSS) Datentypen und Befehle ihre eigenen, gut dokumentierten Leistungssignaturen.

- [Am besten] Eine bewährte Methode besteht darin, das unbeabsichtigte Überschreiben von Daten zu reduzieren. Verwenden Sie eine Namenskonvention, die überlappende Schlüsselnamen

minimiert. Bei der herkömmlichen Benennung Ihrer Datenstrukturen wird eine hierarchische Methode wie APPNAME : CONTEXT : ID bzw. ORDER-APP : CUSTOMER : 123 verwendet.

[Ressourcen]:

- [Benennung von Schlüsseln](#)
- [Best] -Befehle ElastiCache (RedisOSS) haben eine Zeitkomplexität, die durch die Big-O-Notation definiert wird. Hier ist die Komplexität eines Befehls eine algorithmische/mathematische Darstellung seiner Wirkung. Bei der Einführung eines neuen Datentyps in Ihre Anwendung müssen Sie die Zeitkomplexität der zugehörigen Befehle sorgfältig überprüfen. Befehle mit einer Zeitkomplexität von  $O(1)$  sind zeitlich konstant und hängen nicht von der Größe der Eingabe ab. Befehle mit einer Zeitkomplexität von  $O(N)$  sind jedoch zeitlich linear und hängen von der Größe der Eingabe ab. Aufgrund des Single-Thread-Designs von ElastiCache (RedisOSS) führt ein großes Volumen an Vorgängen mit hoher Zeitkomplexität zu einer geringeren Leistung und zu potenziellen Zeitüberschreitungen.

[Ressourcen]:

- [Befehle](#)
- [Am besten] Verwenden Sie diese APIs Option, um sich einen GUI Überblick über das Datenmodell in Ihrem Cluster zu verschaffen.

[Ressourcen]:

- [Redis OSS Commander](#)
- [Redis-Browser OSS](#)
- [Redsmin](#)

## PE 7: Wie protokollieren Sie langsam laufende Befehle in Ihrem ElastiCache Amazon-Cluster?

Einführung auf Fragenebene: Die Leistungsoptimierung bietet Vorteile durch die Erfassung, Aggregation und Benachrichtigung von Befehlen mit langer Laufzeit. Wenn Sie wissen, wie lange die Ausführung von Befehlen dauert, können Sie feststellen, welche Befehle zu einer schlechten Leistung führen und welche Befehle die Engine daran hindern, optimal zu funktionieren. ElastiCache (RedisOSS) kann diese Informationen auch an Amazon CloudWatch oder Amazon Kinesis Data Firehose weiterleiten.

Vorteil auf Fragenebene: Die Protokollierung an einem festen Standort und die Bereitstellung von Benachrichtigungsereignissen bei langsamen Befehlen können zu einer detaillierten Leistungsanalyse beitragen und zur Auslösung automatisierter Ereignisse verwendet werden.

- [Erforderlich] Amazon ElastiCache (RedisOSS) führt Engine Version 6.0 oder neuer aus, ordnungsgemäß konfigurierte Parametergruppe und SLOWLOG Protokollierung sind auf dem Cluster aktiviert.
  - Die erforderlichen Parameter sind nur verfügbar, wenn die Engine-Versionskompatibilität auf Valkey 7.2 und höher oder Redis OSS Version 6.0 oder höher eingestellt ist.
  - SLOWLOGDie Protokollierung erfolgt, wenn die Serverausführungszeit eines Befehls länger dauert als ein bestimmter Wert. Das Verhalten des Clusters hängt von den zugehörigen Parametergruppenparametern ab, d. h. `slowlog-log-slower-than` und `slowlog-max-len`.
  - Änderungen werden sofort wirksam.
- [Am besten] Nutzen Sie unsere Kinesis Data Firehose Firehose-Funktionen. CloudWatch
  - Nutzen Sie die Filter- und Alarmfunktionen von CloudWatch CloudWatch Logs Insights und Amazon Simple Notification Services, um die Leistung zu überwachen und Ereignisse zu benachrichtigen.
  - Verwenden Sie die Streaming-Funktionen von Kinesis Data Firehose, um SLOWLOG Protokolle im permanenten Speicher zu archivieren oder um eine automatische Cluster-Parameteroptimierung auszulösen.
  - Finden Sie heraus, JSON ob TEXT das einfache Format Ihren Anforderungen am besten entspricht.
  - Erteilen Sie IAM Berechtigungen zum Veröffentlichen in CloudWatch oder Kinesis Data Firehose.
- [Besser] Konfigurieren Sie `slowlog-log-slower-than` auf einen anderen Wert als den Standardwert.
  - Dieser Parameter bestimmt, wie lange ein Befehl innerhalb der Valkey- oder OSS Redis-Engine ausgeführt werden kann, bevor er als langsam ausgeführter Befehl protokolliert wird. Der Standardwert ist 10 000 Mikrosekunden (10 Millisekunden). Der Standardwert ist für einige Workloads möglicherweise zu hoch.
  - Ermitteln Sie auf der Grundlage der Anwendungsanforderungen und der Testergebnisse einen Wert, der für Ihren Workload besser geeignet ist. Ein zu niedriger Wert kann jedoch zu übermäßigen Daten führen.
- [Besser] Übernehmen Sie für `slowlog-max-len` den Standardwert.

- Dieser Parameter bestimmt die Obergrenze dafür, wie viele langsam laufende Befehle zu einem bestimmten Zeitpunkt im Valkey- oder OSS Redis-Speicher erfasst werden. Mit dem Wert 0 wird die Erfassung effektiv deaktiviert. Je höher der Wert, desto mehr Einträge werden im Speicher gespeichert. Dadurch wird die Wahrscheinlichkeit verringert, dass wichtige Informationen bereinigt werden, bevor sie überprüft werden können. Der Standardwert lautet 128.
- Der Standardwert ist für die meisten Workloads geeignet. Wenn Daten in einem erweiterten Zeitfenster von der Valkey-CLI aus über den Befehl analysiert werden müssen, sollten Sie erwägen, diesen Wert zu erhöhen. SLOWLOG Dadurch können mehr Befehle im Speicher von Valkey oder Redis verbleiben. OSS

Wenn Sie die SLOWLOG Daten entweder an CloudWatch Logs oder Kinesis Data Firehose weitergeben, werden die Daten dauerhaft gespeichert und können außerhalb des ElastiCache Systems analysiert werden, sodass keine große Anzahl langsam laufender Befehle im Valkey- oder Redis-Speicher gespeichert werden muss. OSS

- [Ressourcen]:
  - [Wie aktiviere ich Slow Log in einem ElastiCache \(Redis-\) Cache-Cluster? OSS](#)
  - [Protokollzustellung](#)
  - [Redis-spezifische Parameter OSS](#)
  - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
  - [Amazon Kinesis Data Firehose](#)

## PE8: Wie hilft Auto Scaling dabei, die Leistung des ElastiCache Clusters zu steigern?

Einführung auf Fragenebene: Durch die Implementierung der Funktion von Valkey oder Redis OSS Auto Scaling können sich Ihre ElastiCache Komponenten im Laufe der Zeit anpassen, um die gewünschten Shards oder Replicas automatisch zu erhöhen oder zu verringern. Dies kann erreicht werden, indem entweder die Zielverfolgung oder die geplante Skalierungsrichtlinie implementiert wird.

Vorteil auf Fragenebene: Wenn Sie die Arbeitslastspitzen verstehen und entsprechend planen, können Sie eine verbesserte Caching-Leistung und einen unterbrechungsfreien Betrieb sicherstellen. ElastiCache (RedisOSS) Auto Scaling überwacht kontinuierlich Ihre CPU /Memory-Auslastung, um sicherzustellen, dass Ihr Cluster mit den gewünschten Leistungsniveaus arbeitet.

- [Erforderlich] Beim Starten eines Clusters für ElastiCache (OSSRedis):
  1. Stellen Sie sicher, dass der Clustermodus aktiviert ist.

2. Stellen Sie sicher, dass die Instance zu einer Familie mit einem bestimmten Typ und einer bestimmten Größe gehört, die Auto Scaling unterstützt.
3. Vergewissern Sie sich, dass der Cluster nicht in globalen Datenspeichern, Outposts oder lokalen Zonen ausgeführt wird

[Ressourcen]:

- [Skalierung von Clustern in Valkey und Redis OSS \(Clustermodus aktiviert\)](#)
- [Verwenden von Auto Scaling mit Shards](#)
- [Verwenden von Auto Scaling mit Replikaten](#)
- [Am besten] Stellen Sie fest, ob Ihr Workload lese- oder schreibintensiv ist, um die Skalierungsrichtlinie zu definieren. Verwenden Sie nur eine Nachverfolgungsmetrik, um eine optimale Leistung zu erzielen. Es wird empfohlen, für jede Dimension mehrere Richtlinien zu vermeiden, da Auto-Scaling-Richtlinien aufskalieren, wenn das Ziel erreicht wird, aber erst dann abskalieren, wenn alle Ziel-Nachverfolgungsrichtlinien für die Abskalierung bereit sind.

[Ressourcen]:

- [Auto-Scaling-Richtlinien](#)
- [Definieren einer Skalierungsrichtlinie](#)
- [Am besten] Wenn Sie die Leistung im Zeitverlauf überwachen, können Sie Workload-Änderungen erkennen, die bei punktueller Überwachung unbemerkt bleiben würden. Sie können die entsprechenden CloudWatch Metriken für die Clusterauslastung über einen Zeitraum von vier Wochen analysieren, um den Zielschwellenwert zu bestimmen. Wenn Sie sich immer noch nicht sicher sind, welchen Wert Sie wählen möchten, empfehlen wir, mit dem minimal unterstützten vordefinierten Metrikwert zu beginnen.

[Ressourcen]:

- [Überwachung der Nutzung mit Metriken CloudWatch](#)
- [Besser] Wir empfehlen, Ihre Anwendung mit den erwarteten minimalen und maximalen Workloads zu testen, um die genaue Anzahl der Shard/Replikate zu ermitteln, die der Cluster benötigt, um Skalierungsrichtlinien zu entwickeln und Verfügbarkeitsprobleme zu minimieren.

[Ressourcen]:

- [Registrieren eines skalierbaren Ziels](#)
- [Registrierung eines skalierbaren Ziels mit dem AWS CLI](#)

# Säule zur ElastiCache Kostenoptimierung bei Amazon Well-Architected Lens

Die Säule der Kostenoptimierung konzentriert sich auf die Vermeidung unnötiger Kosten. Zu den wichtigsten Themen gehören das Verständnis und die Kontrolle, wofür Geld ausgegeben wird, die Auswahl des am besten geeigneten Knotentyps (verwenden Sie Instances, die das Daten-Tiering auf der Grundlage der Workload-Anforderungen unterstützen), die richtige Anzahl von Ressourcentypen (wie viele Lesereplikate), die Analyse der Ausgaben im Zeitverlauf und die Skalierung, um Geschäftsanforderungen zu erfüllen, ohne zu viel Geld auszugeben.

## Themen

- [COST1: Wie identifizieren und verfolgen Sie die mit Ihren ElastiCache Ressourcen verbundenen Kosten? Wie entwickeln Sie Mechanismen, die es Benutzern ermöglichen, Ressourcen zu erstellen, zu verwalten und zu entsorgen?](#)
- [COST2: Wie nutzen Sie Tools zur kontinuierlichen Überwachung, um die mit Ihren ElastiCache Ressourcen verbundenen Kosten zu optimieren?](#)
- [COST3: Sollten Sie einen Instance-Typ verwenden, der Data Tiering unterstützt? Was sind die Vorteile von Daten-Tiering? Wann sollten Daten-Tiering-Instances nicht verwendet werden?](#)

**COST1: Wie identifizieren und verfolgen Sie die mit Ihren ElastiCache Ressourcen verbundenen Kosten? Wie entwickeln Sie Mechanismen, die es Benutzern ermöglichen, Ressourcen zu erstellen, zu verwalten und zu entsorgen?**

Einführung auf Fragenebene: Um Kostenmetriken zu verstehen, müssen mehrere Teams beteiligt werden und zusammenarbeiten: Softwareentwicklung, Datenmanagement, Produktbesitzer, Finanzen und Führung. Um die wichtigsten Kostentreiber zu identifizieren, müssen alle Beteiligten die Hebel der Kontrolle der Servicenutzung und die Kompromisse beim Kostenmanagement verstehen. Dies ist häufig der entscheidende Unterschied zwischen erfolgreichen und weniger erfolgreichen Bemühungen zur Kostenoptimierung. Wenn Sie sicherstellen, dass Sie über Prozesse und Tools verfügen, mit denen Sie die Ressourcen verfolgen können, die von der Entwicklung über die Produktion bis hin zur Außerbetriebnahme entstanden sind, können Sie die damit verbundenen Kosten besser kontrollieren.

Vorteil auf Fragenebene: Die kontinuierliche Erfassung aller mit Ihrem Workload verbundenen Kosten erfordert ein tiefes Verständnis der Architektur, zu der ElastiCache auch eine ihrer Komponenten

gehört. Darüber hinaus sollten Sie über einen Kostenmanagementplan verfügen, um die Nutzung zu erfassen und mit Ihrem Budget zu vergleichen.

- [Erforderlich] Richten Sie ein Cloud Center of Excellence (CCoE) mit einer der Gründungschartas ein, das die Definition, Nachverfolgung und Umsetzung von Kennzahlen rund um die Nutzung in Ihrem Unternehmen selbst übernimmt. ElastiCache Falls eines CCoE existiert und funktioniert, stellen Sie sicher, dass es weiß, wie die damit verbundenen Kosten zu lesen und nachzuverfolgen sind. ElastiCache Wenn Ressourcen erstellt werden, überprüfen Sie anhand von IAM Rollen und Richtlinien, ob nur bestimmte Teams und Gruppen Ressourcen instanzieren können. Damit wird sichergestellt, dass die Kosten mit den Geschäftsergebnissen in Verbindung stehen und dass im Hinblick auf die Kosten eine klare Linienverantwortung festgelegt wird.
  1. CCoE sollte Kostenkennzahlen identifizieren, definieren und veröffentlichen, die regelmäßig — monatlich — aktualisiert werden und sich auf die wichtigsten ElastiCache Nutzungsdaten beziehen, z. B.:
    - a. Verwendete Knotentypen und ihre Attribute: Standard oder speicheroptimiert, On-Demand-Instances oder Reserved Instances, Regionen und Availability Zones
    - b. Arten von Umgebungen: kostenlose Umgebungen, Entwicklungsumgebungen, Testumgebungen und Produktionsumgebungen
    - c. Strategien zur Speicherung und Aufbewahrung von Backups
    - d. Datentransfer innerhalb und zwischen Regionen
    - e. Instances, die in Amazon Outposts ausgeführt werden
  2. CCoE besteht aus einem funktionsübergreifenden Team, in dem nicht ausschließlich die Teams für Softwareentwicklung, Datenmanagement, Produktteam, Finanzen und Führungskräfte in Ihrem Unternehmen vertreten sind.

[Ressourcen]:

- [Schaffen eines Cloud-Kompetenzzentrums](#)
- [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Verwenden Sie Kostenzuordnungs-Tags, um die Kosten auf einer niedrigen Granularitätsstufe zu verfolgen. Verwenden Sie AWS Cost Management, um Ihre AWS Kosten und Nutzung im Laufe der Zeit zu visualisieren, zu verstehen und zu verwalten.
  1. Verwenden Sie Tags, um Ihre Ressourcen zu organisieren, und Tags zur Kostenzuweisung, um Ihre AWS Kosten detailliert zu verfolgen. Nachdem Sie die Kostenzuordnungs-Tags aktiviert haben, AWS verwendet es die Kostenzuordnungs-Tags, um Ihre Ressourcenkosten in Ihrem Kostenzuordnungsbericht zu organisieren, sodass Sie Ihre AWS Kosten leichter kategorisieren

und verfolgen können. AWS stellt zwei Arten von Kostenverrechnungs-Tags bereit: AWS generierte Tags und benutzerdefinierte Tags. AWS definiert, erstellt und wendet die AWS generierten Tags für Sie an, und Sie definieren, erstellen und wenden benutzerdefinierte Tags an. Sie müssen beide Arten von Tags separat aktivieren, bevor sie in AWS-Kostenmanagement oder einem Kostenzuordnungsbericht angezeigt werden können.

2. Verwenden Sie Tags für die Kostenzuweisung, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Wenn Sie Ihren Ressourcen in Amazon Kostenzuordnungs-Tags hinzufügen ElastiCache, können Sie die Kosten verfolgen, indem Sie die Ausgaben auf Ihren Rechnungen nach Ressourcen-Tag-Werten gruppieren. Erwägen Sie, ob Sie Tags miteinander kombinieren, um Kosten detaillierter zu verfolgen.

[Ressourcen]:

- [Verwenden von AWS Kostenzuweisungs-Tags](#)
  - [Überwachen von Kosten mit Kostenzuordnungs-Tags](#)
  - [AWS Cost Explorer](#)
- [Am besten] Verknüpfen Sie ElastiCache Kosten mit Kennzahlen, die sich auf das gesamte Unternehmen auswirken.
    1. Berücksichtigen Sie sowohl Geschäftsmetriken als auch betriebliche Metriken wie die Latenz – welche Konzepte in Ihrem Geschäftsmodell sind rollenübergreifend verständlich? Die Metriken müssen für so viele Rollen wie möglich in der Organisation verständlich sein.
    2. Beispiele: gleichzeitig bereitgestellte Benutzer, maximale und durchschnittliche Latenz pro Operation und Benutzer, Benutzerbindungsergebnisse, Rückkehrraten von Benutzern pro Woche, Sitzungslänge/Benutzer, Abbruchrate, Cache-Trefferrate und verfolgte Schlüssel

[Ressourcen]:

- [Überwachung der Nutzung mit CloudWatch Metriken](#)
- [Gut] Sorgen Sie für einen Überblick über die up-to-date Architektur und den Betrieb der Kennzahlen und Kosten für den gesamten verwendeten Workload ElastiCache.
    1. Verstehen Sie Ihr gesamtes Lösungssystem, das in der ElastiCache Regel Teil eines vollständigen Ökosystems von AWS Diensten in seinem Technologieangebot ist, von Kunden über API Gateway und QuickSight Redshift bis hin zu Reporting-Tools (zum Beispiel).
    2. Tragen Sie die Komponenten Ihrer Lösung wie Clients, Verbindungen, Sicherheit, In-Memory-Betrieb, Speicher, Ressourcenautomatisierung, Datenzugriff und -verwaltung in Ihrem Architekturdiagramm auf. Jede Ebene ist mit der gesamten Lösung verbunden und hat eigene

Anforderungen und Funktionen, welche die Gesamtkosten erhöhen und/oder Ihnen helfen, diese zu verwalten.

3. Ihr Diagramm sollte die Nutzung von Rechenleistung, Netzwerk, Speicher, Lebenszyklusrichtlinien, die Erfassung von Kennzahlen sowie die betrieblichen und funktionalen ElastiCache Elemente Ihrer Anwendung beinhalten
4. Die Anforderungen Ihres Workloads werden sich wahrscheinlich im Laufe der Zeit ändern. Es ist wichtig, dass Sie Ihr Wissen über die zugrundeliegenden Komponenten sowie Ihre primären funktionalen Ziele aufrechterhalten und dokumentieren, um Ihr Workload-Kostenmanagement proaktiv zu gestalten.
5. Die Unterstützung der Geschäftsleitung in Bezug auf Transparenz, Rechenschaftspflicht, Priorisierung und Ressourcen ist entscheidend für eine effektive Kostenmanagementstrategie für Ihr Unternehmen. ElastiCache

**COST2: Wie nutzen Sie Tools zur kontinuierlichen Überwachung, um die mit Ihren ElastiCache Ressourcen verbundenen Kosten zu optimieren?**

Einführung auf Fragenebene: Sie müssen ein ausgewogenes Verhältnis zwischen Ihren ElastiCache Kosten- und Anwendungsleistungskennzahlen anstreben. Amazon CloudWatch bietet Einblicke in wichtige Betriebskennzahlen, anhand derer Sie beurteilen können, ob Ihre ElastiCache Ressourcen im Verhältnis zu Ihren Anforderungen über- oder unterausgelastet sind. Im Hinblick auf die Kostenoptimierung müssen Sie verstehen, wann Ihre Ressourcen überlastet sind, und in der Lage sein, geeignete Mechanismen zu entwickeln, um die Größe Ihrer ElastiCache Ressourcen zu ändern und gleichzeitig Ihre Betriebs-, Verfügbarkeits-, Belastbarkeits- und Leistungsanforderungen aufrechtzuerhalten.

Vorteil auf Fragenebene: Im Idealfall haben Sie genügend Ressourcen bereitgestellt, um die betrieblichen Anforderungen Ihres Workloads zu erfüllen, und verfügen nicht über unterausgelastete Ressourcen, was zu einer suboptimalen Kostensituation führen kann. Sie müssen in der Lage sein, überdimensionierte ElastiCache Ressourcen sowohl zu identifizieren als auch zu vermeiden, sie über einen längeren Zeitraum zu betreiben.

- [Erforderlich] Wird verwendet CloudWatch , um Ihre ElastiCache Cluster zu überwachen und zu analysieren, wie sich diese Metriken auf Ihre AWS Cost Explorer Explorer-Dashboards beziehen.
  1. ElastiCache bietet sowohl Metriken auf Host-Ebene (z. B. CPU Nutzung) als auch Metriken, die für die Cache-Engine-Software spezifisch sind (z. B. Cache-Abrufe und Cache-Fehlschläge).

Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.

2. ElastiCache Leistungskennzahlen (CPUUtilization,, EngineUtilizationSwapUsage, und Räumungen) können darauf hinweisen CurrConnections, dass Sie nach oben oder unten (mit größeren/kleineren Cache-Knotentypen) oder nach innen oder außen (mehr/weniger Shards hinzufügen) skalieren müssen. Machen Sie sich mit den Auswirkungen von Skalierungsentscheidungen auf die Kosten vertraut, indem Sie eine Playbook-Matrix erstellen, in der die zusätzlichen Kosten sowie die Mindest- und Höchstdauer geschätzt werden, die erforderlich sind, um die Schwellenwerte für die Anwendungsleistung zu erreichen.

[Ressourcen]:

- [Überwachung der Nutzung mithilfe von Metriken CloudWatch](#)
  - [Welche Metriken sollte ich überwachen?](#)
  - [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Verstehen und dokumentieren Sie Ihre Backup-Strategie und die Auswirkungen auf die Kosten.
    1. Mit ElastiCache, die Backups werden in Amazon S3 gespeichert, was eine dauerhafte Speicherung bietet. Sie müssen die Kostenauswirkungen verstehen, die sich aus Ihrer Fähigkeit ergeben, sich nach Ausfällen zu erholen.
    2. Aktivieren Sie automatische Backups, mit denen Backup-Dateien gelöscht werden, die das Aufbewahrungslimit überschritten haben.

[Ressourcen]:

- [Planen automatischer Backups](#)
  - [Amazon Simple Storage Service – Preise](#)
- [Am besten] Verwenden Sie reservierte Knoten für Ihre Instances als bewusste Strategie zur Kostenkontrolle für Workloads, die wohlverstanden und dokumentiert sind. Für reservierte Knoten wird eine Vorausgebühr erhoben, die von der Art des Knotens und der Dauer der Reservierung – ein oder drei Jahre – abhängt. Diese Gebühr ist wesentlich geringer als die stündliche Nutzungsgebühr, die bei On-Demand-Knoten anfällt.
    1. Möglicherweise müssen Sie Ihre ElastiCache Cluster mit On-Demand-Knoten betreiben, bis Sie genügend Daten gesammelt haben, um die Reserved Instance-Anforderungen abzuschätzen. Planen und dokumentieren Sie die Ressourcen, die Sie benötigen, um Ihre Anforderungen zu erfüllen, und vergleichen Sie die erwarteten Kosten für die verschiedenen Instance-Typen ([On-Demand-Instances und Reserved Instances](#)).

2. Evaluieren Sie regelmäßig die neuen verfügbaren Cache-Knotentypen und beurteilen Sie, ob es aus Sicht der Kosten- und Betriebsmetriken sinnvoll ist, Ihre Instance-Flotte auf neue Cache-Knotentypen zu migrieren.

**COST3: Sollten Sie einen Instance-Typ verwenden, der Data Tiering unterstützt?**

**Was sind die Vorteile von Daten-Tiering? Wann sollten Daten-Tiering-Instances nicht verwendet werden?**

Einführung auf Fragenebene: Die Auswahl des geeigneten Instance-Typs wirkt sich nicht nur auf die Leistung und das Service-Level aus, sondern hat auch finanzielle Auswirkungen. Mit Instance-Typen sind unterschiedliche Kosten verbunden. Die Auswahl eines oder mehrerer großer Instance-Typen, die alle Speicheranforderungen im Arbeitsspeicher erfüllen, könnte sich anbieten. Dies könnte jedoch erhebliche Kostenauswirkungen haben, sobald das Projekt ausgereift ist. Um sicherzustellen, dass der richtige Instance-Typ ausgewählt ist, muss die Leerlaufzeit von ElastiCache Objekten regelmäßig überprüft werden.

Vorteil auf Fragenebene: Sie sollten genau wissen, wie sich verschiedene Instance-Typen derzeit und in Zukunft auf Ihre Kosten auswirken. Geringe oder regelmäßige Workload-Änderungen sollten nicht zu unverhältnismäßigen Kostenänderungen führen. Wenn der Workload es zulässt, bieten Instance-Typen, die Daten-Tiering unterstützen, einen besseren Preis pro verfügbarem Speicher. Aufgrund der pro Instanz verfügbaren SSD Speicherdaten unterstützen Tiering-Instances eine wesentlich höhere Gesamtdatenmenge pro Instanz.

- [Erforderlich] Kennen der Einschränkungen von Daten-Tiering-Instances
  1. Nur für ElastiCache (Redis-OSS) Cluster verfügbar.
  2. Nur begrenzte Instance-Typen unterstützen Daten-Tiering.
  3. Nur ElastiCache (RedisOSS) Version 6.2 und höher wird unterstützt
  4. Große Artikel werden nicht ausgetauscht. SSD Objekte über 128 MiB werden im Speicher aufbewahrt.

[Ressourcen]:

- [Daten-Tiering](#)
- [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Finden Sie heraus, auf welchen Anteil Ihrer Datenbank in Prozent Ihr Workload regelmäßig zugreift.

1. Daten-Tiering-Instances eignen sich für Workloads, die häufig auf einen kleinen Teil Ihres gesamten Datensatzes zugreifen, aber dennoch einen schnellen Zugriff auf die verbleibenden Daten benötigen. Mit anderen Worten, das Verhältnis von sehr häufig zu häufig abgerufenen Daten beträgt etwa 20:80.
  2. Entwickeln Sie eine Nachverfolgung der Leerlaufzeit von Objekten auf Clusterebene.
  3. Große Implementierungen mit über 500 GB Daten sind gute Kandidaten.
- [Erforderlich] Machen Sie sich bewusst, dass Daten-Tiering-Instances für bestimmte Workloads nicht optional sind.
    1. Der Zugriff auf weniger häufig verwendete Objekte verursacht geringe Leistungseinbußen, da diese auf lokale SSD Objekte ausgelagert werden. Wenn Ihre Anwendung empfindlich auf die Reaktionszeit reagiert, testen Sie die Auswirkungen auf Ihren Workload.
    2. Nicht geeignet für Caches, die hauptsächlich große Objekte mit einer Größe von über 128 MiB speichern.

[Ressourcen]:

- [Einschränkungen](#)
- [Am besten] Reserved-Instance-Typen unterstützen Daten-Tiering. Dadurch werden die niedrigsten Kosten in Bezug auf die Menge an Datenspeicher pro Instance gewährleistet.
  1. Möglicherweise müssen Sie Ihre ElastiCache Cluster mit Instances ohne Datenklassifizierung betreiben, bis Sie Ihre Anforderungen besser verstanden haben.
  2. Analysieren Sie das ElastiCache Datennutzungsmuster Ihres Clusters.
  3. Erstellen Sie einen automatisierten Auftrag, der in regelmäßigen Abständen die Leerlaufzeit von Objekten erfasst.
  4. Wenn Sie feststellen, dass ein großer Prozentsatz (etwa 80 %) der Objekte für einen Zeitraum inaktiv ist, der für Ihren Workload als angemessen erachtet wird, dokumentieren Sie die Ergebnisse und schlagen Sie vor, den Cluster auf Instances zu migrieren, die Daten-Tiering unterstützen.
  5. Evaluieren Sie regelmäßig die neuen verfügbaren Cache-Knotentypen und beurteilen Sie, ob es aus Sicht der Kosten- und Betriebsmetriken sinnvoll ist, Ihre Instance-Flotte auf neue Cache-Knotentypen zu migrieren.

[Ressourcen]:

- [OBJECT IDLETIME](#)
- [ElastiCacheAmazon-Preisgestaltung](#)

# Allgemeine Schritte zur Fehlerbehebung und bewährte Methoden mit ElastiCache

Die folgenden Themen enthalten Hinweise zur Fehlerbehebung bei Fehlern und Problemen, die bei der Verwendung auftreten können. ElastiCache Wenn Sie ein Problem finden, das hier nicht aufgeführt ist, können Sie es über die Feedback-Schaltfläche auf dieser Seite melden.

Weitere Tipps zur Fehlerbehebung und Antworten auf häufig gestellte Supportfragen finden Sie im [AWS Knowledge Center](#)

## Themen

- [Verbindungsprobleme](#)
- [Valkey- oder Redis-Client-Fehler OSS](#)
- [Fehlerbehebung bei hoher Latenz in Serverless ElastiCache](#)
- [Behebung von Drosselungsproblemen in Serverless ElastiCache](#)
- [Anhaltende Verbindungsprobleme](#)
- [Verwandte Themen](#)

## Verbindungsprobleme

Wenn Sie keine Verbindung zu Ihrem ElastiCache Cache herstellen können, ziehen Sie eine der folgenden Möglichkeiten in Betracht:

1. Verwendung TLS: Wenn beim Versuch, eine Verbindung zu Ihrem ElastiCache Endpunkt herzustellen, ein Verbindungsproblem auftritt, verwenden Sie es möglicherweise nicht TLS in Ihrem Client. Wenn Sie ElastiCache Serverless verwenden, ist die Verschlüsselung bei der Übertragung immer aktiviert. Stellen Sie sicher, dass Ihr Client TLS die Verbindung zum Cache verwendet. [Erfahren Sie mehr über das Herstellen einer Verbindung zu einem TLS aktivierten Cache.](#)
2. VPC: Auf ElastiCache Caches kann nur von einem VPC aus zugegriffen werden. Stellen Sie sicher, dass die EC2 Instanz, von der aus Sie auf den Cache zugreifen, und der ElastiCache Cache in derselben VPC Instanz erstellt wurden. Alternativ müssen Sie das [VPC Peering](#) zwischen dem Ort aktivieren, VPC an dem sich Ihre EC2 Instance befindet, und dem VPC Ort, an dem Sie Ihren Cache erstellen.

3. Sicherheitsgruppen: ElastiCache verwendet Sicherheitsgruppen, um den Zugriff auf Ihren Cache zu kontrollieren. Berücksichtigen Sie dabei Folgendes:
  - a. Stellen Sie sicher, dass die von Ihrem ElastiCache Cache verwendete Sicherheitsgruppe eingehenden Zugriff von Ihrer EC2 Instance aus darauf zulässt. [Hier](#) erfahren Sie, wie Sie Regeln für eingehenden Datenverkehr in Ihrer Sicherheitsgruppe korrekt einrichten.
  - b. Stellen Sie sicher, dass die von Ihrem ElastiCache Cache verwendete Sicherheitsgruppe den Zugriff auf die Ports Ihres Caches ermöglicht (6379 und 6380 für serverlose Verbindungen und standardmäßig 6379 für selbst entworfene Ports). ElastiCache verwendet diese Ports, um Valkey- oder Redis-Befehle zu akzeptieren. OSS [Erfahren Sie hier mehr darüber, wie Sie den Portzugriff einrichten.](#)

Wenn die Verbindung weiterhin schwierig ist, finden Sie [Anhaltende Verbindungsprobleme](#) weitere Schritte.

## Valkey- oder Redis-Client-Fehler OSS

ElastiCache Auf Serverless kann nur über Clients zugegriffen werden, die das Valkey- oder OSS Redis-Clustermodus-Protokoll unterstützen. Auf selbst entworfene Cluster kann von Clients in beiden Modi zugegriffen werden, abhängig von der Clusterkonfiguration.

Wenn bei Ihrem Client Fehler auftreten, sollten Sie Folgendes beachten:

1. Clustermodus: Wenn bei dem [SELECT](#)Befehl CROSSLOT Fehler oder Fehler auftreten, versuchen Sie möglicherweise, mit einem Valkey- oder OSS Redis-Client, der das Cluster-Protokoll nicht unterstützt, auf einen Cache mit aktiviertem Clustermodus zuzugreifen. ElastiCache Serverless unterstützt nur Clients, die das Valkey- oder Redis-Clusterprotokoll unterstützen. OSS Wenn Sie Valkey oder Redis OSS im „Cluster-Modus deaktiviert“ (CMD) verwenden möchten, müssen Sie Ihren eigenen Cluster entwerfen.
2. CROSSLOT-Fehler: Wenn der ERR CROSSLOT Keys in request don't hash to the same slot Fehler auftritt, versuchen Sie möglicherweise, auf Schlüssel zuzugreifen, die nicht zu demselben Steckplatz in einem Clustermodus-Cache gehören. Zur Erinnerung: ElastiCache Serverless arbeitet immer im Clustermodus. Operationen mit mehreren Schlüsseln, Transaktionen oder Lua-Skripten mit mehreren Schlüsseln sind nur zulässig, wenn sich alle beteiligten Schlüssel im selben Hash-Slot befinden.

[Weitere bewährte Methoden zur Konfiguration von Valkey- oder OSS Redis-Clients finden Sie in diesem Blogbeitrag.](#)

## Fehlerbehebung bei hoher Latenz in Serverless ElastiCache

Wenn bei Ihrer Workload eine hohe Latenz auftritt, können Sie anhand der `SuccessfulWriteRequestLatency` Messwerte CloudWatch `SuccessfulReadRequestLatency` und überprüfen, ob die Latenz mit ElastiCache Serverless zusammenhängt. Diese Metriken messen die Latenz, die innerhalb von ElastiCache Serverless liegt. Die clientseitige Latenz und die Netzwerkausfallzeiten zwischen Ihrem Client und dem ElastiCache serverlosen Endpunkt sind nicht enthalten.

### Fehlerbehebung bei der clientseitigen Latenz

Wenn Sie eine erhöhte Latenz auf der Clientseite, aber keinen entsprechenden Anstieg CloudWatch `SuccessfulReadRequestLatency` der `SuccessfulWriteRequestLatency` Messwerte für die serverseitige Latenz feststellen, sollten Sie Folgendes beachten:

- Stellen Sie sicher, dass die Sicherheitsgruppe den Zugriff auf die Ports 6379 und 6380 zulässt: ElastiCache Serverless verwendet den 6379-Port für den primären Endpunkt und den 6380-Port für den Leser-Endpunkt. Einige Clients stellen für jede neue Verbindung eine Verbindung zu beiden Ports her, auch wenn Ihre Anwendung die Funktion „Aus Replikat lesen“ nicht verwendet. Wenn Ihre Sicherheitsgruppe keinen eingehenden Zugriff auf beide Ports zulässt, kann der Verbindungsaufbau länger dauern. Erfahren Sie [hier](#) mehr darüber, wie Sie den Portzugriff einrichten.

### Fehlerbehebung bei serverseitiger Latenz

Einige Schwankungen und gelegentliche Spitzenwerte sollten keinen Anlass zur Sorge geben. Wenn die Average Statistik jedoch einen starken Anstieg zeigt und anhält, sollten Sie im AWS Health Dashboard und in Ihrem Personal Health Dashboard nach weiteren Informationen suchen. Falls erforderlich, erwägen Sie, einen Support-Fall mit zu eröffnen. AWS Support

Ziehen Sie die folgenden bewährten Methoden und Strategien zur Reduzierung der Latenz in Betracht:

- Read from Replica aktivieren: Wenn Ihre Anwendung dies zulässt, empfehlen wir, die Funktion „Read from Replica“ in Ihrem Valkey- oder OSS Redis-Client zu aktivieren, um Lesevorgänge zu skalieren und eine geringere Latenz zu erreichen. Wenn diese Option aktiviert ist, versucht ElastiCache Serverless, Ihre Leseanfragen an Replikat-Cache-Knoten weiterzuleiten, die sich in derselben Availability Zone (AZ) wie Ihr Client befinden, wodurch AZ-übergreifende Netzwerklatenzen vermieden werden. Beachten Sie, dass die Aktivierung der Funktion „Aus

Replikat lesen“ in Ihrem Client bedeutet, dass Ihre Anwendung eine eventuelle Datenkonsistenz akzeptiert. Ihre Anwendung empfängt möglicherweise für einige Zeit ältere Daten, wenn Sie versuchen, sie zu lesen, nachdem Sie in einen Schlüssel geschrieben haben.

- Stellen Sie sicher, dass Ihre Anwendung im selben AZs Cache bereitgestellt wird: Möglicherweise stellen Sie eine höhere clientseitige Latenz fest, wenn Ihre Anwendung nicht im selben AZs Cache bereitgestellt wird. Wenn Sie einen serverlosen Cache erstellen, können Sie die Subnetze angeben, von denen aus Ihre Anwendung auf den Cache zugreift, und ElastiCache Serverless erstellt VPC Endpoints in diesen Subnetzen. Stellen Sie sicher, dass Ihre Anwendung in derselben Umgebung bereitgestellt wird. AZs Andernfalls kann es bei Ihrer Anwendung beim Zugriff auf den Cache zu einem AZ-übergreifenden Hop kommen, was zu einer höheren clientseitigen Latenz führt.
- Verbindungen wiederverwenden: ElastiCache Serverlose Anfragen werden über eine TLS aktivierte TCP Verbindung mithilfe des Protokolls gestellt. RESP Das Initiieren der Verbindung (einschließlich der Authentifizierung der Verbindung, falls konfiguriert) nimmt Zeit in Anspruch, sodass die Latenz der ersten Anfrage höher als üblich ist. Anfragen über eine bereits initialisierte Verbindung sorgen für eine gleichbleibend niedrige ElastiCache Latenz. Aus diesem Grund sollten Sie die Verwendung von Verbindungspooling oder die Wiederverwendung vorhandener Valkey- oder Redis-Verbindungen in Betracht ziehen. OSS
- Skalierungsgeschwindigkeit: ElastiCache Serverless skaliert automatisch, wenn Ihre Anforderungsrate steigt. Ein plötzlicher starker Anstieg der Anforderungsrate, der schneller ist als die Geschwindigkeit, mit der ElastiCache Serverless skaliert, kann für einige Zeit zu einer erhöhten Latenz führen. ElastiCache Serverless kann die unterstützte Anforderungsrate in der Regel schnell erhöhen. Es dauert bis zu 10 bis 12 Minuten, bis sich die Anforderungsrate verdoppelt.
- Untersuchen Sie Befehle mit langer Laufzeit: Einige Valkey- oder OSS Redis-Befehle, einschließlich Lua-Skripten oder Befehle für große Datenstrukturen, können lange laufen. ElastiCache Veröffentlicht Metriken auf Befehlsebene, um diese Befehle zu identifizieren. Mit [ElastiCache Serverless](#) können Sie die BasedECPUs Metriken verwenden.
- Gedrosselte Anfragen: Wenn Anfragen in ElastiCache Serverless gedrosselt werden, kann es zu einem Anstieg der clientseitigen Latenz in Ihrer Anwendung kommen. [Wenn Anfragen in Serverless gedrosselt werden, sollten Sie einen ElastiCache Anstieg der Serverless-Metrik feststellen. ThrottledRequests ElastiCache](#) Im folgenden Abschnitt finden Sie Informationen zur Behebung gedrosselter Anfragen.
- Gleichmäßige Verteilung von Schlüsseln und Anfragen: ElastiCache Bei Valkey und Redis OSS kann eine ungleichmäßige Verteilung von Schlüsseln oder Anfragen pro Steckplatz zu einem Hot-Slot führen, was zu einer erhöhten Latenz führen kann. ElastiCache Serverless unterstützt bis

zu 30.000 ECPUs /Sekunde (90.000 ECPUs /Sekunde bei Verwendung von Read from Replica) auf einem einzigen Steckplatz in einer Arbeitslast, die einfache /-Befehle ausführt. SET GET Wir empfehlen, Ihre Schlüssel- und Anforderungsverteilung auf die einzelnen Slots zu überprüfen und für eine gleichmäßige Verteilung zu sorgen, falls Ihre Anforderungsrate diese Grenze überschreitet.

## Behebung von Drosselungsproblemen in Serverless ElastiCache

In serviceorientierten Architekturen und verteilten Systemen wird die Begrenzung der Geschwindigkeit, mit der API Anrufe von verschiedenen Servicekomponenten verarbeitet werden, als Drosselung bezeichnet. Dadurch werden Spannungsspitzen ausgeglichen, Diskrepanzen im Komponentendurchsatz kontrolliert und bei unerwarteten Betriebsereignissen besser vorhersehbare Wiederherstellungen ermöglicht. ElastiCache Serverless ist für diese Art von Architekturen konzipiert, und die meisten Valkey- oder Redis-Clients verfügen über integrierte OSS Wiederholungsversuche für gedrosselte Anfragen. Ein gewisses Maß an Drosselung ist nicht zwangsläufig ein Problem für Ihre Anwendung. Eine anhaltende Drosselung eines latenzsensitiven Teils des Datenworkflows kann sich jedoch negativ auf die Benutzererfahrung auswirken und die allgemeine Effizienz des Systems beeinträchtigen.

[Wenn Anfragen in Serverless gedrosselt werden, sollten Sie einen Anstieg der ElastiCache Serverless-Metrik feststellen. ThrottledRequests ElastiCache](#) Wenn Sie eine hohe Anzahl gedrosselter Anfragen feststellen, sollten Sie Folgendes beachten:

- **Skalierungsgeschwindigkeit:** ElastiCache Serverless wird automatisch skaliert, wenn Sie mehr Daten aufnehmen oder Ihre Anforderungsrate erhöhen. Wenn Ihre Anwendung schneller skaliert als die Geschwindigkeit, mit der Serverless skaliert, werden Ihre Anfragen möglicherweise gedrosselt, während ElastiCache Serverless an Ihre Arbeitslast angepasst wird. ElastiCache ElastiCache Serverless kann die Speichergröße in der Regel schnell erhöhen. Es dauert bis zu 10 bis 12 Minuten, bis sich die Speichergröße in Ihrem Cache verdoppelt hat.
- **Gleichmäßige Verteilung von Schlüsseln und Anfragen:** ElastiCache Bei Valkey oder Redis OSS kann eine ungleichmäßige Verteilung von Schlüsseln oder Anfragen pro Steckplatz zu einem Hot-Slot führen. Ein Hot-Slot kann zu einer Drosselung von Anfragen führen, wenn die Anforderungsrate für einen einzelnen Slot 30.000 ECPUs /Sekunde übersteigt, bei einer Arbeitslast, die einfache /-Befehle ausführt. SET GET
- **Aus Replikat lesen:** Wenn Ihre Anwendung dies zulässt, sollten Sie in Erwägung ziehen, die Funktion „Aus Replikat lesen“ zu verwenden. Die meisten Valkey- oder OSS Redis-Clients können so konfiguriert werden, dass sie Lesevorgänge skalieren „, sodass Lesevorgänge direkt an Replikatknoten weitergeleitet werden. Mit dieser Funktion können Sie den Lesetraffic skalieren.

Darüber hinaus leitet ElastiCache Serverless automatisch Lesevorgänge von Replikatanfragen an Knoten weiter, die sich in derselben Availability Zone wie Ihre Anwendung befinden, was zu einer geringeren Latenz führt. Wenn Read from Replica aktiviert ist, können Sie bei Workloads mit einfachen /-Befehlen bis zu 90.000 ECPUs /Sekunde an einem einzigen Steckplatz erreichen. SET  
GET

## Anhaltende Verbindungsprobleme

Bei der Behebung anhaltender Verbindungsprobleme müssen die folgenden Punkte überprüft werden ElastiCache:

### Themen

- [Sicherheitsgruppen](#)
- [Netzwerk ACLs](#)
- [Routing-Tabellen](#)
- [DNSLösung](#)
- [Identifizieren von Problemen mit serverseitiger Diagnose](#)
- [Netzwerkverbindbarkeit](#)
- [Netzwerkbezogene Grenzen](#)
- [CPUVerwendung](#)
- [Verbindungen, die von der Serverseite beendet werden](#)
- [Clientseitige Fehlerbehebung für Amazon-Instances EC2](#)
- [Aufschlüsselung der Zeit, die zum Abschließen einer einzelnen Anfrage benötigt wird](#)

## Sicherheitsgruppen

Sicherheitsgruppen sind virtuelle Firewalls, die Ihren ElastiCache Client (EC2Instance, AWS Lambda Funktion, ECS Amazon-Container usw.) und Ihren ElastiCache Cache schützen. Sicherheitsgruppen sind statusbehaftet, was bedeutet, dass nach dem Zulassen des eingehenden oder ausgehenden Datenverkehrs die Antworten für diesen Datenverkehr automatisch im Kontext dieser bestimmten Sicherheitsgruppe autorisiert werden.

Die Statusfunktion erfordert, dass die Sicherheitsgruppe alle autorisierten Verbindungen verfolgt, und es gibt ein Limit für verfolgte Verbindungen. Wenn das Limit erreicht ist, schlagen neue Verbindungen

fehl. Im Abschnitt zur Fehlerbehebung finden Sie Hilfe dazu, wie Sie feststellen können, ob die Grenzwerte auf dem Client oder auf der ElastiCache Seite erreicht wurden.

Sie können dem Client und dem ElastiCache Cluster eine einzelne Sicherheitsgruppe gleichzeitig oder einzelne Sicherheitsgruppen für jede Gruppe zuweisen.

In beiden Fällen müssen Sie den TCP ausgehenden Verkehr auf dem ElastiCache Port von der Quelle und den eingehenden Verkehr auf demselben Port zulassen. ElastiCache Der Standardport ist 11211 für Memcached und 6379 für Valkey oder Redis. OSS Standardmäßig gestatten Sicherheitsgruppen allen ausgehenden Datenverkehr. In diesem Fall ist nur die eingehende Regel in der Zielsicherheitsgruppe erforderlich.

Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cluster in einem Amazon VPC](#).

## Netzwerk ACLs

Network Access Control Lists (ACLs) sind statuslose Regeln. Der Datenverkehr muss in beide Richtungen (Eingehend und Ausgehend) zugelassen sein, um erfolgreich zu sein. Netzwerke ACLs werden Subnetzen zugewiesen, nicht bestimmten Ressourcen. Es ist möglich, dass dieselben Ressourcen ElastiCache und die Client-Ressource ACL zugewiesen werden, insbesondere, wenn sie sich im selben Subnetz befinden.

Standardmäßig lässt das Netzwerk den gesamten ACLs Datenverkehr zu. Es ist jedoch möglich, sie anzupassen, um Datenverkehr zu verweigern oder zu erlauben. Darüber hinaus erfolgt die Auswertung der ACL Regeln sequentiell, was bedeutet, dass die Regel, deren niedrigste Zahl dem Verkehr entspricht, sie zulässt oder verweigert. Die Mindestkonfiguration, um den Valkey- oder OSS Redis-Verkehr zuzulassen, ist:

### Clientseitiges Netzwerk: ACL

- Regeln für eingehenden Datenverkehr:
- Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
- Typ: Benutzerdefinierte TCP Regel;
- Protokoll: TCP
- Portbereich: 1024-65535
- Quelle: 0.0.0.0/0 (oder erstellen Sie individuelle Regeln für die Cluster-Subnetze) ElastiCache
- Erlauben/Verweigern

- Regeln für ausgehenden Datenverkehr:
- Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
- TypTCP: Benutzerdefinierte Regel;
- Protokoll: TCP
- Portbereich: 6379
- Quelle: 0.0.0.0/0 (oder die Cluster-Subnetze. ElastiCache Beachten Sie, dass die Verwendung bestimmter Optionen im Falle eines Failovers oder der Skalierung des Clusters zu Problemen führen IPs kann.
- Erlauben/Verweigern

#### ElastiCache NetzwerkACL:

- Regeln für eingehenden Datenverkehr:
  - Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
  - Typ: Benutzerdefinierte TCP Regel;
  - Protokoll: TCP
  - Portbereich: 6379
  - Quelle: 0.0.0.0/0 (oder erstellen Sie individuelle Regeln für die Cluster-Subnetze) ElastiCache
  - Erlauben/Verweigern
- 
- Regeln für ausgehenden Datenverkehr:
  - Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
  - TypTCP: Benutzerdefinierte Regel;
  - Protokoll: TCP
  - Portbereich: 1024-65535
  - Quelle: 0.0.0.0/0 (oder die Cluster-Subnetze. ElastiCache Beachten Sie, dass die Verwendung bestimmter Optionen im Falle eines Failovers oder der Skalierung des Clusters zu Problemen führen IPs kann.
  - Erlauben/Verweigern

Weitere Informationen finden Sie unter [Netzwerk ACLs](#).

## Routing-Tabellen

Ähnlich wie beim Netzwerk ACLs kann jedes Subnetz unterschiedliche Routing-Tabellen haben. Wenn sich Clients und der ElastiCache Cluster in unterschiedlichen Subnetzen befinden, stellen Sie sicher, dass ihre Routing-Tabellen es ihnen ermöglichen, einander zu erreichen.

Bei komplexeren Umgebungen mit mehreren VPCs dynamischen Routing- oder Netzwerk-Firewalls kann es schwierig werden, Fehler zu beheben. Siehe [Netzwerkverbindbarkeit](#), um zu bestätigen, dass Ihre Netzwerkeinstellungen angemessen sind.

## DNSLösung

ElastiCache stellt die Dienstendpunkte auf der Grundlage von DNS Namen bereit. Die verfügbaren Endpunkte sind `Configuration`, `Primary`, `Reader`, und `Node`-Endpunkte. Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#).

Im Falle eines Failovers oder einer Clusteränderung kann sich die dem Endpunktnamen zugeordnete Adresse ändern und wird automatisch aktualisiert.

Benutzerdefinierte DNS Einstellungen (d. h. wenn der VPC DNS Dienst nicht verwendet wird) kennen die von ElastiCache -bereitgestellten DNS Namen möglicherweise nicht. Stellen Sie sicher, dass Ihr System die ElastiCache Endpunkte mithilfe von Systemtools wie `dig` (wie unten gezeigt) oder erfolgreich auflösen kann. `nslookup`

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
1.2.3.4
```

Sie können die Namensauflösung auch über den VPC DNS Dienst erzwingen:

```
$ 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
```

## Identifizieren von Problemen mit serverseitiger Diagnose

CloudWatch Metriken und Laufzeitinformationen der ElastiCache Engine sind gängige Informationsquellen, um mögliche Ursachen für Verbindungsprobleme zu identifizieren. Eine gute Analyse beginnt üblicherweise mit den folgenden Elementen:

- **CPU-Verwendung:** Valkey und Redis OSS sind Multithread-Anwendungen. Die Ausführung jedes Befehls erfolgt jedoch in einem einzigen (Haupt-) Thread. Aus diesem Grund bietet ElastiCache die Metriken `EngineCPUUtilization` und `CPUUtilization`. `EngineCPUUtilization` bietet die CPU-Nutzung, die für den Valkey- oder OSS Redis-Prozess vorgesehen ist, und `CPUUtilization` die Nutzung für alle vCPUs. Knoten mit mehr als einem vCPU haben normalerweise unterschiedliche Werte für `EngineCPUUtilization` und `CPUUtilization`, wobei der zweite Wert in der Regel höher ist. Ein hoher Wert `EngineCPUUtilization` kann durch eine erhöhte Anzahl von Anfragen oder durch komplexe Vorgänge verursacht werden, deren Ausführung viel CPU-Zeit in Anspruch nimmt. Sie können beide folgendermaßen identifizieren:
  - **Erhöhte Anzahl von Anforderungen:** Prüfen Sie auf Erhöhungen für andere Metriken, die den `EngineCPUUtilization`-Muster. Nützliche Metriken sind:
    - **CacheHits und CacheMisses:** Die Anzahl der erfolgreichen Anforderungen oder Anforderungen, die kein gültiges Element im Cache gefunden haben. Wenn das Verhältnis von Fehlern im Vergleich zu Treffern hoch ist, verschwendet die Anwendung Zeit und Ressourcen mit unfruchtbaren Anfragen.
    - **SetTypeCmds und GetTypeCmds:** Diese Metriken, die mit `EngineCPUUtilization` korreliert sind, können helfen zu verstehen, ob die Last für Schreibanforderungen signifikant höher ist, gemessen durch `SetTypeCmds`, oder liest, gemessen durch `GetTypeCmds`. Wenn die Last überwiegend Lesevorgänge ist, kann die Verwendung mehrerer Read-Replicas die Anforderungen über mehrere Knoten hinweg ausgleichen und die primäre für Schreibvorgänge ersparen. In Clustern mit deaktiviertem Clustermodus können Read-Replicas verwendet werden, indem in der Anwendung mithilfe des Reader-Endpunkts eine zusätzliche Konfiguration erstellt wird. ElastiCache Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#). Die Lesevorgänge müssen an diese zusätzliche Verbindung gesendet werden. Schreibvorgänge werden über den regulären primären Endpunkt durchgeführt. Im Clustermodus ist es ratsam, eine Bibliothek zu verwenden, die Read-Replicas nativ unterstützt. Mit den richtigen Flags kann die Bibliothek automatisch die Cluster-Topologie und die Replikatknoten erkennen, die Lesevorgänge über den OSS-Befehl `READONLY` Valkey oder Redis aktivieren und die Leseanfragen an die Replikate senden.
  - **Erhöhte Anzahl von Verbindungen:**
    - **CurrConnections und NewConnections:** `CurrConnections` ist die Anzahl der etablierten Verbindungen zum Zeitpunkt der Datenpunkt-Sammlung, während `NewConnections` zeigt an, wie viele Verbindungen in der Periode erstellt wurden.

Das Erstellen und Verarbeiten von Verbindungen ist mit erheblichem Aufwand verbunden. CPU Darüber hinaus wirkt sich der TCP Drei-Wege-Handshake, der zum Herstellen neuer Verbindungen erforderlich ist, negativ auf die Gesamtantwortzeiten aus.

Ein ElastiCache Knoten mit Tausenden `NewConnections` pro Minute bedeutet, dass eine Verbindung mit nur wenigen Befehlen hergestellt und verwendet wird, was nicht optimal ist. Es ist eine bewährte Vorgehensweise, Verbindungen herzustellen und sie für neue Vorgänge wiederzuverwenden. Dies ist möglich, wenn die Clientanwendung Verbindungspooling oder persistente Verbindungen unterstützt und ordnungsgemäß implementiert. Beim Verbindungspooling wird die Anzahl der `currConnections` hat keine großen Variationen, und die `NewConnections` sollte so niedrig wie möglich sein. Valkey und Redis OSS bieten optimale Leistung bei einer geringen Anzahl von `currConnections`. Wenn Sie die `currConnection` Reihenfolge von zehn oder hundert einhalten, wird der Ressourcenverbrauch für die Unterstützung einzelner Verbindungen wie Client-Puffer und CPU Zyklen zur Bereitstellung der Verbindung minimiert.

- **Netzwerkdurchsatz:**
  - Ermitteln Sie die Bandbreite: Die Netzwerkbandbreite der ElastiCache Knoten ist proportional zur Knotengröße. Da Anwendungen unterschiedliche Merkmale aufweisen, können die Ergebnisse je nach Workload variieren. Beispielsweise wirken sich Anwendungen mit einer hohen Anzahl kleiner Anfragen tendenziell stärker auf die CPU Nutzung als auf den Netzwerkdurchsatz aus, während größere Schlüssel zu einer höheren Netzwerkauslastung führen. Aus diesem Grund ist es ratsam, die Knoten mit der tatsächlichen Workload zu testen, um die Grenzen besser zu verstehen.

Die Simulation der Last aus der Anwendung würde genauere Ergebnisse liefern. Benchmark-Tools können jedoch eine gute Vorstellung von den Grenzen geben.

- In Fällen, in denen die Anforderungen überwiegend Lesevorgänge sind, verringert die Verwendung von Replikaten für Lesevorgänge die Belastung des primären Knotens. Wenn der Anwendungsfall überwiegend Schreibvorgänge ist, wird die Verwendung vieler Replikate die Netzwerknutzung verstärken. Für jedes Byte, das auf den primären Knoten geschrieben wird, werden  $N$  Bytes an die Replikate gesendet, wobei  $N$  die Anzahl der Replikate ist. Die bewährte Methode für schreibintensive Workloads ist die Verwendung von ElastiCache (RedisOSS) mit aktiviertem Clustermodus, sodass die Schreibvorgänge auf mehrere Shards verteilt werden können, oder die Skalierung auf einen Knotentyp mit mehr Netzwerkfunktionen.

- Die CloudWatchmetrics `NetworkBytesIn` und `NetworkBytesOut` geben die Datenmenge an, die in den Knoten eingeht bzw. den Knoten verlässt. `ReplicationBytes` ist der Verkehr, der der Datenreplikation gewidmet ist.

Weitere Informationen finden Sie unter [Netzwerkbezogene Grenzen](#).

- Komplexe Befehle: OSS Redis-Befehle werden in einem einzigen Thread bedient, was bedeutet, dass Anfragen sequentiell bedient werden. Ein einzelner langsamer Befehl kann sich auf andere Anforderungen und Verbindungen auswirken, was zu Timeouts führt. Die Verwendung von Befehlen, die auf mehrere Werte, Schlüssel oder Datentypen wirken, muss sorgfältig durchgeführt werden. Verbindungen können abhängig von der Anzahl der Parameter oder der Größe der Ein- oder Ausgabewerte blockiert oder beendet werden.

Ein berüchtigtes Beispiel ist die `KEYS`-Befehl. Es fegt den gesamten Schlüsselraum auf der Suche nach einem bestimmten Muster und blockiert die Ausführung anderer Befehle während der Ausführung. Redis OSS verwendet die Notation „Big O“, um die Komplexität seiner Befehle zu beschreiben.

Keys Befehl hat  $O(N)$  Zeitkomplexität, wobei  $N$  die Anzahl der Schlüssel in der Datenbank ist. Je größer die Anzahl der Schlüssel ist, desto langsamer wird der Befehl. `KEYS` kann auf verschiedene Arten Probleme verursachen: Wenn kein Suchmuster verwendet wird, gibt der Befehl alle verfügbaren Schlüsselnamen zurück. In Datenbanken mit tausend oder Millionen von Elementen wird eine riesige Ausgabe erstellt und die Netzwerkpuffer überflutet.

Wenn ein Suchmuster verwendet wird, werden nur die Schlüssel, die dem Muster entsprechen, an den Client zurückgegeben. Die Engine wird jedoch immer noch den gesamten Schlüsselraum durchsucht, und die Zeit, um den Befehl abzuschließen, ist gleich.

Eine Alternative für `KEYS` ist die `SCAN`-Befehl. Es iteriert über den Schlüsselraum und begrenzt die Iterationen in einer bestimmten Anzahl von Elementen, wodurch längere Blöcke auf der Engine vermieden werden.

Der Scan hat die `COUNT`, der zum Festlegen der Größe der Iterationsblöcke verwendet wird. Der Standardwert ist 10 (10 Elemente pro Iteration).

Abhängig von der Anzahl der Elemente in der Datenbank sind kleine `COUNT`-Werte-Blöcke erfordern mehr Iterationen, um einen vollständigen Scan abzuschließen, und größere Werte halten die Engine bei jeder Iteration länger beschäftigt. Während kleine Zählwerte `SCAN`

langsamer in großen Datenbanken machen, können größere Werte die gleichen Probleme verursachen, die für KEYS beschrieben sind.

Als Beispiel wird das Ausführen des SCAN-Befehl mit Zählwert als 10 erfordert 100.000 Wiederholungen in einer Datenbank mit 1 Million Schlüsseln. Wenn die durchschnittliche Netzwerk-Roundtrip-Zeit 0,5 Millisekunden beträgt, werden etwa 50.000 Millisekunden (50 Sekunden) für die Übertragung von Anforderungen ausgegeben.

Auf der anderen Seite, wenn der Zählwert 100,000 wäre, wäre eine einzelne Iteration erforderlich und nur 0,5 ms würden ausgegeben, um sie zu übertragen. Die Engine wäre jedoch für andere Operationen vollständig blockiert, bis der Befehl den gesamten Schlüsselraum beendet hat.

Außerdem KEYS, sind einige andere Befehle potenziell schädlich, wenn sie nicht korrekt verwendet werden. Eine Liste aller Befehle und ihrer jeweiligen Zeitkomplexität finden Sie unter [Valkey- und OSS Redis-Befehle](#).

Beispiele für mögliche Probleme:

- Lua-Skripte: Valkey und Redis OSS bieten einen eingebetteten Lua-Interpreter, der die Ausführung von Skripten auf der Serverseite ermöglicht. Lua-Skripte auf Valkey und Redis OSS werden auf Engine-Ebene ausgeführt und sind per Definition atomar, was bedeutet, dass kein anderer Befehl oder Skript ausgeführt werden darf, während ein Skript ausgeführt wird. Lua-Skripte bieten die Möglichkeit, mehrere Befehle, Entscheidungsalgorithmen, Datenanalyse und andere direkt auf der Engine auszuführen. Während die Atomizität von Skripten und die Möglichkeit, die Anwendung zu entladen, verlockend sind, müssen Skripte mit Sorgfalt und für kleine Operationen verwendet werden. Bei ElastiCache aktivierter Option ist die Ausführungszeit von Lua-Skripten auf 5 Sekunden begrenzt. Skripte, die nicht in den Schlüsselraum geschrieben wurden, werden nach Ablauf der 5 Sekunden automatisch beendet. Um Datenbeschädigungen und Inkonsistenzen zu vermeiden, wird der Knoten ein Failover ausgeführt, wenn die Skriptausführung nicht innerhalb von 5 Sekunden abgeschlossen wurde und während der Ausführung Schreibvorgänge stattfand. [Transaktionen](#) sind die Alternative, um die Konsistenz mehrerer verwandter wichtiger Änderungen in OSS Redis zu gewährleisten. Eine Transaktion ermöglicht die Ausführung eines Blocks von Befehlen und überwacht vorhandene Schlüssel auf Änderungen. Wenn sich einer der überwachten Schlüssel vor Abschluss der Transaktion ändert, werden alle Änderungen verworfen.

- **Massenlöschung von Elementen:** Die `DEL` akzeptiert mehrere Parameter, bei denen es sich um die Schlüsselnamen handelt, die gelöscht werden sollen. Löschvorgänge sind synchron und nehmen CPU viel Zeit in Anspruch, wenn die Liste der Parameter umfangreich ist oder eine große Liste, einen Satz, eine sortierte Menge oder einen Hash (Datenstrukturen mit mehreren Unterelementen) enthält. Mit anderen Worten, selbst das Löschen eines einzelnen Schlüssels kann erhebliche Zeit in Anspruch nehmen, wenn es viele Elemente enthält. Die Alternative dazu `DEL` ist ein asynchroner Befehl `UNLINK`, der seit Redis 4 verfügbar ist. `UNLINK` muss, `DEL` wann immer möglich, vorgezogen werden. Ab ElastiCache (RedisOSS) 6.x sorgt der `lazyfree-lazy-user-del` Parameter dafür, dass sich der `DEL` Befehl so verhält, `UNLINK` als ob er aktiviert wäre. Weitere Informationen finden Sie unter [Redis OSS 6.0-Parameteränderungen](#).
- **Befehle, die auf mehrere Tasten wirken:** `DEL` wurde zuvor als Befehl erwähnt, der mehrere Argumente akzeptiert und seine Ausführungszeit wird direkt proportional dazu sein. Redis OSS bietet jedoch viele weitere Befehle, die ähnlich funktionieren. Als Beispiele `MSET` und `MGET` ermöglichen das gleichzeitige Einfügen oder Abrufen mehrerer String-Schlüssel. Ihre Nutzung kann vorteilhaft sein, um die Netzwerklatenz zu reduzieren, die mehreren einzelnen `SET` oder `GET`-Befehle. Eine umfangreiche Liste von Parametern wirkt sich jedoch auf die CPU Verwendung aus.

Die CPU Auslastung allein ist zwar nicht die Ursache für Verbindungsprobleme, aber wenn zu viel Zeit für die Verarbeitung einzelner oder weniger Befehle über mehrere Schlüssel aufgewendet wird, kann dies dazu führen, dass andere Anfragen fehlschlagen und die CPU Gesamtauslastung erhöht wird.

Die Anzahl der Schlüssel und ihre Größe beeinflussen die Komplexität des Befehls und damit die Fertigstellungszeit.

Weitere Beispiele für Befehle, die auf mehrere Tasten wirken können: `HMGET`, `HMSET`, `MSETNX`, `PFCOUNT`, `PFMERGE`, `SDIFF`, `SDIFFSTORE`, `SINTER`, `SINTERSTORE`, `SUNION`, `SUNIONSTORE`, `TOUCH`, `ZDIFF`, `ZDIFFSTORE`, `ZINTER` oder `ZINTERSTORE`.

- **Befehle, die auf mehrere Datentypen einwirken:** Redis bietet OSS auch Befehle, die auf einen oder mehrere Schlüssel wirken, unabhängig von ihrem Datentyp. ElastiCache (RedisOSS) stellt die Metrik `KeyBasedCmds` zur Überwachung solcher Befehle bereit. Diese Metrik summiert die Ausführung der folgenden Befehle im ausgewählten Zeitraum:
  - Komplexität von  $O(N)$ :
    - `KEYS`

- O(1)
  - EXISTS
  - OBJECT
  - PTTL
  - RANDOMKEY
  - TTL
  - TYPE
  - EXPIRE
  - EXPIREAT
  - MOVE
  - PERSIST
  - PEXPIRE
  - PEXPIREAT
  - UNLINK (O(N), um Speicher zurückzugewinnen. Die Speicherwiederherstellungsaufgabe geschieht jedoch in einem getrennten Thread und blockiert die Engine nicht
- Unterschiedliche Komplexitätszeiten je nach Datentyp:
  - DEL
  - DUMP
  - RENAME wird als Befehl mit O(1)-Komplexität betrachtet, führt aber DEL intern. Die Ausführungszeit hängt von der Größe des umbenannten Schlüssels ab.
  - RENAMENX
  - RESTORE
  - SORT
- Große Hashes: Hash ist ein Datentyp, der einen einzelnen Schlüssel mit mehreren Schlüssel-Wert-Unterelementen erlaubt. Jeder Hash kann 4.294.967.295 Elemente speichern und Operationen auf großen Hashes können teuer werden. Ähnlich wie KEYS, haben Hashes die HKEYS Befehl mit O(N) Zeitkomplexität, wobei N die Anzahl der Elemente im Hash ist. HSCAN hat Vorrang vor HKEYS, um lange laufende Befehle zu vermeiden. HDEL, HGETALL, HMGET, HMSET und HVALS sind Befehle, die bei großen Hashes mit Vorsicht verwendet werden sollten.

- Andere Big-Data-Strukturen: Neben Hashes können auch andere Datenstrukturen intensiv sein. CPU Auch Sets, Listen, Sortierte Sets und Hyperloglogs können abhängig von ihrer Größe und den verwendeten Befehlen viel Zeit in Anspruch nehmen. Weitere Informationen zu diesen Befehlen finden Sie unter [Valkey- und Redis-Befehle](#). OSS

## Netzwerkonnektivität

Nach der Überprüfung der Netzwerkkonfigurationen in Bezug auf DNS Auflösung, SicherheitsgruppenACLs, Netzwerk- und Routingtabellen kann die Konnektivität mit dem VPC Reachability Analyzer und den Systemtools überprüft werden.

Reachability Analyzer testet die Netzwerkkonnektivität und bestätigt, ob alle Anforderungen und Berechtigungen erfüllt sind. Für die folgenden Tests benötigen Sie die ENI ID (Elastic Network Interface Identification) eines der in Ihrem System verfügbaren ElastiCache Knoten. VPC Gehen Sie dazu wie folgt vor:

1. Gehen Sie zu <https://console.aws.amazon.com/ec2/v2/home? #: NIC>
2. Filtern Sie die Schnittstellenliste nach Ihrem ElastiCache Clusternamen oder der IP-Adresse, die Sie zuvor bei den DNS Validierungen erhalten haben.
3. Notieren Sie sich die ID oder speichern Sie sie auf andere Weise. ENI Wenn mehrere Schnittstellen angezeigt werden, überprüfen Sie die Beschreibung, um sicherzustellen, dass sie zum richtigen ElastiCache Cluster gehören, und wählen Sie eine davon aus.
4. Fahren Sie mit dem nächsten Schritt fort.
5. Zu <https://console.aws.amazon.com/vpc/Hause einen Analysepfad erstellen? #> ReachabilityAnalyzer und wählen Sie die folgenden Optionen:
  - Quelltyp: Wählen Sie die Instance, wenn Ihr ElastiCache Client auf einer EC2 Amazon-Instance läuft, oder eine Netzwerkschnittstelle (falls er einen anderen Service verwendet, z. B. AWS Fargate Amazon ECS mit awsvpc-Netzwerk usw.), und die entsprechende Ressourcen-ID (EC2Instance oder ENI ID); AWS Lambda
  - Zieltyp: Wählen Sie Network Interface und wählen Sie den Elasticache ENI aus der Liste aus.
  - Zielport: Geben Sie 6379 für ElastiCache (RedisOSS) oder 11211 für ElastiCache (Memcached) an. Dies sind die Ports, die mit der Standardkonfiguration definiert sind, und in diesem Beispiel wird davon ausgegangen, dass sie nicht geändert werden.
  - Protokoll: TCP

Erstellen Sie den Analyse-Pfad und warten Sie ein paar Augenblicke auf das Ergebnis. Wenn der Status nicht erreichbar ist, öffnen Sie die Analysedetails und überprüfen Sie die Analyse-Explorer für Details, in denen die Anfragen blockiert wurden.

Wenn die Erreichbarkeitstests bestanden haben, fahren Sie mit der Überprüfung auf Betriebssystemebene fort.

Um die TCP Konnektivität auf dem ElastiCache Service-Port zu überprüfen: Auf Amazon Linux Nping ist es im Paket enthalten nmap und kann die TCP Konnektivität auf dem ElastiCache Port testen sowie die Netzwerk-Round-Trip-Zeit für den Verbindungsaufbau angeben. Verwenden Sie diese Option, um die Netzwerkkonnektivität und die aktuelle Latenz zum ElastiCache Cluster zu überprüfen, wie im Folgenden dargestellt:

```
$ 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
```

In der Standardeinstellung sendet Nping 5 Sonden mit einer Verzögerung von 1 Sekunde zwischen ihnen. Sie können die Option „-c“ verwenden, um die Anzahl der Prüfpunkte zu erhöhen und „—delay“, um die Zeit für das Senden eines neuen Tests zu ändern.

Wenn die Tests mit nping fehlschlagen und die VPC Reachability Analyzer-Tests bestanden haben, bitten Sie Ihren Systemadministrator, mögliche hostbasierte Firewallregeln, asymmetrische Routing-Regeln oder andere mögliche Einschränkungen auf Betriebssystemebene zu überprüfen.

Überprüfen Sie auf der ElastiCache Konsole, ob die Verschlüsselung während der Übertragung in Ihren Clusterdetails aktiviert ist. ElastiCache Wenn die Verschlüsselung bei der Übertragung aktiviert ist, überprüfen Sie mit dem folgenden Befehl, ob die TLS Sitzung eingerichtet werden kann:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Wenn die Verbindung und die TLS Verhandlung erfolgreich sind, wird eine umfangreiche Ausgabe erwartet. Überprüfen Sie den in der letzten Zeile verfügbaren Rückgabecode, der Wert muss 0 (ok)

sein. Wenn openssl etwas anderes zurückgibt, überprüfen Sie den Grund für den Fehler unter <https://www.openssl.org/docs/man1.0.2/man1/verify.html#.DIAGNOSTICS>

Wenn alle Infrastruktur- und Betriebssystemtests bestanden wurden, Ihre Anwendung aber immer noch keine Verbindung herstellen kann ElastiCache, überprüfen Sie, ob die Anwendungskonfigurationen den ElastiCache Einstellungen entsprechen. Häufige Fehler sind:

- Ihre Anwendung unterstützt den ElastiCache Clustermodus nicht und der Clustermodus ElastiCache ist aktiviert;
- Ihre Anwendung unterstützt TLS/SSL nicht und die Verschlüsselung bei der Übertragung ElastiCache ist aktiviert;
- Die Anwendung unterstützt TLS/SSL, verfügt aber nicht über die richtigen Konfigurationsflags oder vertrauenswürdigen Zertifizierungsstellen;

## Netzwerkbezogene Grenzen

- Maximale Anzahl von Verbindungen: Es gibt harte Grenzen für gleichzeitige Verbindungen. Jeder ElastiCache Knoten ermöglicht bis zu 65.000 gleichzeitige Verbindungen zwischen allen Clients. Dieses Limit kann mithilfe der eingeschalteten `CurConnections` Metriken überwacht werden. CloudWatch Clients haben jedoch auch ihre Grenzen für ausgehende Verbindungen. Überprüfen Sie unter Linux den zulässigen flüchtigen Portbereich mit folgendem Befehl:

```
sysctl net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 32768 60999
```

Im vorherigen Beispiel sind 28231 Verbindungen von derselben Quelle zu derselben Ziel-IP (ElastiCache Knoten) und demselben Zielport zulässig. Der folgende Befehl zeigt, wie viele Verbindungen für einen bestimmten ElastiCache Knoten (IP 1.2.3.4) bestehen:

```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
'^State' | wc -l
```

Wenn die Zahl zu hoch ist, wird Ihr System möglicherweise überlastet und versucht, die Verbindungsanforderungen zu verarbeiten. Es ist ratsam, Techniken wie Verbindungspooling oder persistente Verbindungen zu implementieren, um die Verbindungen besser zu handhaben. Wenn möglich, konfigurieren Sie den Verbindungspool so, dass die maximale Anzahl von Verbindungen auf einige hundert begrenzt wird. Außerdem wäre eine Back-Off-Logik zur Behandlung von

Timeouts oder anderen Verbindungsausnahmen ratsam, um im Falle von Problemen eine Verbindungsabwanderung zu vermeiden.

- Grenzwerte für den Netzwerkverkehr: Überprüfen Sie die folgenden [CloudWatch Metriken für Redis OSS](#), um mögliche Netzwerklimits zu identifizieren, die auf dem Knoten erreicht werden: ElastiCache
  - `NetworkBandwidthInAllowanceExceeded/NetworkBandwidthOutAllowanceExceeded`: Netzwerkpakete geformt, weil der Durchsatz das aggregierte Bandbreitenlimit überschritten hat.

Es ist wichtig zu beachten, dass jedes Byte, das auf den primären Knoten geschrieben wird, auf N Replikate repliziert wird, wobei N die Anzahl der Replikate ist. Cluster mit kleinen Knotentypen, mehreren Replikaten und intensiven Schreib Anforderungen können den Replikationsrückstand möglicherweise nicht bewältigen. In solchen Fällen ist es eine bewährte Methode, hochzuskalieren (Knoten-Typ ändern), aufzuskalieren (Shards in Cluster mit aktiviertem Cluster-Modus hinzufügen), die Anzahl der Replikate zu reduzieren oder die Anzahl der Schreibvorgänge zu minimieren.

- `NetworkConntrackAllowanceExceeded`: Pakete geformt, weil die maximale Anzahl von Verbindungen, die über alle dem Knoten zugewiesenen Sicherheitsgruppen nachverfolgt werden, überschritten wurde. Neue Verbindungen werden in diesem Zeitraum wahrscheinlich fehlschlagen.
- `NetworkPackets PerSecondAllowanceExceeded`: Maximale Anzahl von Paketen pro Sekunde überschritten. Arbeitslasten, die auf einer hohen Rate von sehr kleinen Anforderungen basieren, können diese Grenze vor der maximalen Bandbreite erreichen.

Die oben genannten Metriken sind der ideale Weg, um zu bestätigen, dass Knoten ihre Netzwerklimits erreichen. Grenzwerte können jedoch auch von Plateaus in Netzwerkmetriken identifiziert werden.

Wenn diese Plateaus über einen längeren Zeitraum beobachtet werden, wird es wahrscheinlich zu Verzögerungen bei der Replikation, einer Zunahme der für den Cache verwendeten Byte, einem Rückgang des freien Speichers, einem hohen Auslagerungsgrad und hoher Auslastung kommen. CPU EC2Amazon-Instances haben auch Netzwerklimits, die anhand von [ENATreibermetriken](#) verfolgt werden können. Linux-Instances mit erweiterter Netzwerkunterstützung und ENA Treibern 2.2.10 oder neuer können die Limit-Zähler mit dem folgenden Befehl überprüfen:

```
ethtool -S eth0 | grep "allowance_exceeded"
```

## CPUVerwendung

Die CPU Nutzungsmetrik ist der Ausgangspunkt für die Untersuchung, und die folgenden Punkte können dabei helfen, mögliche Probleme ElastiCache einzugrenzen:

- Redis OSS SlowLogs: In der ElastiCache Standardkonfiguration werden die letzten 128 Befehle beibehalten, deren Ausführung mehr als 10 Millisekunden gedauert hat. Der Verlauf der langsamen Befehle wird während der Motorlaufzeit beibehalten und geht im Falle eines Fehlers oder eines Neustarts verloren. Wenn die Liste 128 Einträge erreicht, werden alte Ereignisse entfernt, um Raum für neue zu öffnen. Die Größe der Liste der langsamen Ereignisse und die Ausführungszeit, die als langsam betrachtet wird, kann durch die Parameter `slowlog-max-len` und `slowlog-log-slower-than` in einer [Benutzerdefinierte Parametergruppe](#) angepasst werden. Die slowlogs-Liste kann abgerufen werden, indem `SLOWLOG GET 128` auf dem Motor, 128 sind die letzten 128 langsamen Befehle gemeldet. Jeder Spielereintrag hat folgende Felder:

```
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
```

Das oben genannte Ereignis ereignete sich am 26. Dezember um 19:26:07 UTC Uhr. Es dauerte 4,8 Sekunden (4,823 ms) und wurde durch den vom Client angeforderten Befehl 1.2.3.4 verursacht. KEYS

Unter Linux kann der Zeitstempel mit dem Befehlsdatum konvertiert werden:

```
$ date --date='@1609010767'
Sat Dec 26 19:26:07 UTC 2020
```

Mit Python:

```
>>> from datetime import datetime
>>> datetime.fromtimestamp(1609010767)
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

Oder PowerShell unter Windows mit:

```
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
```

Viele langsame Befehle in kurzer Zeit (gleiche Minute oder weniger) sind ein Grund zur Besorgnis. Überprüfen Sie die Art der Befehle und wie sie optimiert werden können (siehe vorangegangene Beispiele). Wenn häufig Befehle mit  $O(1)$ -Zeitkomplexität gemeldet werden, überprüfen Sie die anderen oben genannten Faktoren für eine hohe CPU Auslastung.

- Latenzmetriken: ElastiCache (RedisOSS) bietet CloudWatch Metriken zur Überwachung der durchschnittlichen Latenz für verschiedene Klassen von Befehlen. Der Datenpunkt wird berechnet, indem die Gesamtzahl der Ausführungen von Befehlen in der Kategorie durch die gesamte Ausführungszeit in der Periode dividiert wird. Es ist wichtig zu verstehen, dass Latenzmetrikergebnisse ein Aggregat mehrerer Befehle sind. Ein einzelner Befehl kann unerwartete Ergebnisse wie Timeouts verursachen, ohne signifikante Auswirkungen auf die Metriken zu haben. In solchen Fällen wären die Slowlog-Ereignisse eine genauere Informationsquelle. Die folgende Liste enthält die verfügbaren Latenzmetriken und die entsprechenden Befehle, die sie betreffen.
  - EvalBasedCmdsLatency: bezieht sich auf Lua-Script-Befehle,; eval evalsha
  - GeoSpatialBasedCmdsLatency: geodist, geohash, geopos, georadius, georadiusbymember, geoad;
  - GetTypeCmdsLatency: Befehle lesen, unabhängig vom Datentyp;

- **HashBasedCmdsLatency:** hexists, hget, hgetall, hkeys, hlen, hmget, hvals, hstrlen, hdel, hincrby, hincrbyfloat, hmset, hset, hsetnx;
- **HyperLogLogBasedCmdsLatency:** pfselftest, pfcount, pfdebug, pfadd, pfmerge;
- **KeyBasedCmdsLatency:** Befehle, die auf verschiedene Datentypen einwirken können: dump, exists, keys, object, pttl, randomkey, ttl, type, del, expire, expireat, move, per
- **ListBasedCmdsLatency:** lindex, len, lrange, blpop, brpop, brpoplpush, linsert, lpop, push, pushx, lrem, let, ltrim, rpop, rpoplpush, rpush, rpushx;
- **PubSubBasedCmdsLatency:** abonnieren, veröffentlichen, pubsub, abbestellen, abonnieren, abbestellen;
- **SetBasedCmdsLatency:** scard, sdiff, sinter, sismember, smembers, srandmember, union, sadd, sdiffstore, sinterstore, smove, spop, srem, sunionstore;
- **SetTypeCmdsLatency:** Befehle schreiben, unabhängig vom Datentyp;
- **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;
- **OSSRedis-Laufzeitbefehle:**
  - **info commandstats:** Stellt eine Liste der Befehle bereit, die seit dem Start der OSS Redis-Engine ausgeführt wurden, ihre kumulative Anzahl an Ausführungen, die Gesamtausführungszeit und die durchschnittliche Ausführungszeit pro Befehl;
  - **Client-Liste:** Bietet eine Liste der aktuell verbundenen Clients und relevante Informationen wie Pufferverwendung, zuletzt ausgeführter Befehl usw.;
- **Backup und Replikation:** ElastiCache (Redis-OSS) Versionen vor 2.8.22 verwenden einen Fork-Prozess, um Backups zu erstellen und vollständige Synchronisationen mit den Replikaten durchzuführen. Diese Methode kann in erheblichem Speicheraufwand für schreibintensive Anwendungsfälle auftreten.

Beginnend mit ElastiCache Redis OSS 2.8.22 wurde eine Forkless-Backup- und Replikationsmethode eingeführt. AWS Die neue Methode kann Schreibvorgänge verzögern, um Fehler zu vermeiden. Beide Methoden können zu Zeiten höherer CPU Auslastung, zu längeren Antwortzeiten und folglich zu Timeouts beim Client während ihrer Ausführung führen. Überprüfen Sie immer, ob die Client-Fehler während des Backup-Fensters oder der `SaveInProgress`-Metrik war 1 in der Periode. Es ist ratsam, das Backup-Fenster für Zeiten geringer Auslastung zu planen, um die Möglichkeit von Problemen mit Clients oder Backup-Fehlern zu minimieren.

## Verbindungen, die von der Serverseite beendet werden

Die Standardkonfiguration ElastiCache (RedisOSS) sorgt dafür, dass die Client-Verbindungen auf unbestimmte Zeit hergestellt werden. In einigen Fällen kann eine Verbindungsbeendigung jedoch wünschenswert sein. Zum Beispiel:

- Fehler in der Client-Anwendung können dazu führen, dass Verbindungen vergessen und im Leerlauf gehalten werden. Dies wird als „Verbindungsleck“ bezeichnet und die Folge ist eine stetige Zunahme der Anzahl etablierter Verbindungen, die auf der `CurrentConnections`-Metrik beobachtet werden. Dieses Verhalten kann zu einer Überlastung des Clients oder ElastiCache der Seite führen. Wenn eine sofortige Korrektur von der Client-Seite aus nicht möglich ist, legen einige Administratoren in ihrer ElastiCache Parametergruppe einen Wert für das „Timeout“ fest. Das Timeout ist die Zeit in Sekunden, die erlaubt ist, dass Verbindungen im Leerlauf bestehen bleiben. Wenn der Client innerhalb dieses Zeitraums keine Anfrage einreicht, beendet die OSS Redis-Engine die Verbindung, sobald die Verbindung den Timeout-Wert erreicht. Kleine Zeitüberschreitungswerte können zu unnötigen Trennungen führen, und Clients müssen sie ordnungsgemäß behandeln und eine erneute Verbindung herstellen, was zu Verzögerungen führt.
- Der Speicher, der zum Speichern von Schlüsseln verwendet wird, wird für Clientpuffer freigegeben. Langsame Clients mit großen Anfragen oder Antworten benötigen möglicherweise eine beträchtliche Menge an Speicher, um ihre Puffer zu verarbeiten. Die Standardkonfigurationen ElastiCache (RedisOSS) schränken die Größe der regulären Client-Ausgabepuffer nicht ein. Wenn das `SymbolMemoryLimit` erreicht wird, versucht die Engine, Elemente zu vertreiben, um die Pufferverwendung zu erfüllen. Bei extrem wenig Arbeitsspeicher entscheidet sich ElastiCache (RedisOSS) möglicherweise dafür, die Verbindung von Clients zu trennen, die große Client-Ausgabepuffer verbrauchen, um Speicherplatz freizugeben und den Zustand des Clusters zu erhalten.

Es ist möglich, die Größe von Clientpuffern mit benutzerdefinierten Konfigurationen zu begrenzen, und Clients, die das Limit erreichen, werden getrennt. Clients sollten jedoch in der Lage sein, unerwartete Trennungen zu verarbeiten. Die Parameter für die Puffergröße für reguläre Clients sind die folgenden:

- `client-query-buffer-limit`: Maximale Größe einer einzelnen Eingabeanforderung;
- `client-output-buffer-limit-normal-soft-limit`: Soft-Limit für Client-Verbindungen. Die Verbindung wird beendet, wenn sie länger als die für `client-output-buffer-limit` - definierte Zeit in Sekunden über dem Soft-Limit bleibt `normal-soft-seconds` oder wenn sie das Hard-Limit erreicht;
- `client-output-buffer-limit-normal-soft-seconds`: Zulässige Zeit für Verbindungen, die den Wert von `client-output-buffer-limit` - überschreiten `normal-soft-limit`;
- `client-output-buffer-limit-normal-hard-limit`: Eine Verbindung, die dieses Limit erreicht, wird sofort beendet.

Neben den regulären Clientpuffern steuern die folgenden Optionen den Puffer für Replikatknoten und Pub/Sub (Publish/Sub) Clients:

- `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`;

## Clientseitige Fehlerbehebung für Amazon-Instances EC2

Die Auslastung und Reaktionsfähigkeit auf der Client-Seite können sich auch auf die Anfragen auswirken. ElastiCache EC2 Bei der Behebung von zeitweiligen Verbindungs- oder Timeout-Problemen müssen die Beschränkungen für Instanzen und Betriebssysteme sorgfältig geprüft werden. Einige wichtige Punkte zu beachten:

- CPU:
  - EC2CPUNutzung der Instance: Stellen Sie sicher, dass CPU die Auslastung nicht erreicht wurde oder fast zu 100 Prozent erreicht ist. Die historische Analyse kann wie folgt durchgeführt werden. Beachten Sie jedoch CloudWatch, dass die Granularität der Datenpunkte entweder 1 Minute (bei aktivierter detaillierter Überwachung) oder 5 Minuten beträgt;

- Wenn Sie [EC2Burstable-Instances](#) verwenden, stellen Sie sicher, dass ihr CPU Guthaben nicht aufgebraucht ist. Diese Informationen sind in der Metrik verfügbar. CPUcreditBalance CloudWatch
- Kurze Perioden mit hoher CPU Auslastung können zu Timeouts führen, ohne dass eine 100-prozentige Auslastung berücksichtigt wird. CloudWatch Solche Fälle erfordern eine Echtzeitüberwachung mit Betriebssystem-Tools wie top, ps und mpstat.
- Netzwerk
  - Überprüfen Sie, ob der Netzwerkdurchsatz gemäß den Instance-Funktionen unter akzeptablen Werten liegt. Weitere Informationen finden Sie unter [EC2Amazon-Instance-Typen](#)
  - Bei Instances mit Erweiterten ena-Netzwerktreiber, überprüfen Sie die [EN-Statistiken](#) für Timeouts oder überschritten Limits. Die folgenden Statistiken sind nützlich, um die Sättigung der Netzwerklimits zu bestätigen:
    - `bw_in_allowance_exceeded/bw_out_allowance_exceeded`: Anzahl der Pakete, die durch übermäßigen eingehenden oder ausgehenden Durchsatz geformt werden;
    - `conntrack_allowance_exceeded`: Anzahl der Pakete, die aufgrund von Sicherheitsgruppen [Verbindungsverfolgung von Grenzen](#) gelöscht wurden. Neue Verbindungen werden fehlschlagen, wenn diese Grenze gesättigt ist;
    - `linklocal_allowance_exceeded`: Anzahl der Pakete, die aufgrund übermäßiger Anfragen an Instance-Metadaten verworfen wurden, NTP via. VPC DNS Das Limit beträgt 1024 Pakete pro Sekunde für alle Dienste;
    - `pps_allowance_exceeded`: Anzahl der Pakete, die aufgrund übermäßiger Pakete pro Sekunde verworfen wurden. Das PPS Limit kann erreicht werden, wenn der Netzwerkverkehr aus Tausenden oder Millionen sehr kleiner Anfragen pro Sekunde besteht. ElastiCache Der Datenverkehr kann optimiert werden, um Netzwerkpakete über Pipelines oder Befehle, die mehrere Operationen gleichzeitig ausführen, besser zu nutzen, MGET anstatt GET

## Aufschlüsselung der Zeit, die zum Abschließen einer einzelnen Anfrage benötigt wird

- Im Netzwerk sind: Tcpdump und Wireshark (tshark auf der Befehlszeile) praktische Tools, um zu verstehen, wie viel Zeit die Anfrage benötigt hat, um das Netzwerk zu erreichen, die ElastiCache Engine zu starten und eine Antwort zu erhalten. Im folgenden Beispiel wird eine einzelne Anforderung hervorgehoben, die mit dem folgenden Befehl erstellt wurde:

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
```

```
+PONG
```

Parallel zum obigen Befehl wurde tcpdump ausgeführt und zurückgegeben:

```
$ 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
```

Anhand der obigen Ausgabe können wir bestätigen, dass der TCP Drei-Wege-Handshake in 222 Mikrosekunden (918091 — 917869) abgeschlossen wurde und der Ping-Befehl innerhalb von 173 Mikrosekunden (918295 — 918122) gesendet und zurückgegeben wurde.

Es dauerte 438 Mikrosekunden (918307 - 917869) vom Anfordern bis zum Schließen der Verbindung. Diese Ergebnisse würden bestätigen, dass Netz- und Triebwerksreaktionszeiten gut sind und sich die Untersuchung auf andere Komponenten konzentrieren kann.

- Auf dem Betriebssystem: `strace` kann helfen, Zeitlücken auf Betriebssystemebene zu identifizieren. Die Analyse der tatsächlichen Anwendungen wäre viel umfangreicher und spezialisierte Anwendungsprofiler oder Debugger sind ratsam. Das folgende Beispiel zeigt nur, ob die Basisbetriebssystemkomponenten wie erwartet funktionieren, andernfalls können weitere Untersuchungen erforderlich sein. Wenn OSS PING `strace` wir denselben Redis-Befehl verwenden, erhalten wir:

```
$ 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\0\notls20201224\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\0\notls20201224\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 +++
```

Im obigen Beispiel dauerte der Befehl etwas mehr als 54 Millisekunden (752110 - 697712 = 54398 Mikrosekunden).

Für die Instanziierung von nc und die Namensauflösung (von 697712 bis 717890) wurde eine beträchtliche Zeit, etwa 20 ms, benötigt. Danach waren 2 ms erforderlich, um den TCP Socket zu erstellen (745659 bis 747858) und 0,4 ms (747858 bis 748330), um die Antwort auf die Anfrage zu senden und zu empfangen.

## Verwandte Themen

- [the section called “Bewährte Methoden und Caching-Strategien”](#)

# Sicherheit bei Amazon ElastiCache

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS -Compliance-Programme](#) regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für Amazon gelten ElastiCache, finden Sie unter [AWS Services in Scope by Compliance Program](#).
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung von Amazon anwenden können ElastiCache. In den folgenden Themen erfahren Sie, wie Sie Amazon konfigurieren ElastiCache, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Ihnen helfen, Ihre ElastiCache Amazon-Ressourcen zu überwachen und zu sichern.

## Themen

- [Datenschutz bei Amazon ElastiCache](#)
- [Richtlinie für den Datenverkehr zwischen Netzwerken](#)
- [Identity and Access Management für Amazon ElastiCache](#)
- [Konformitätsvalidierung für Amazon ElastiCache](#)
- [Resilienz bei Amazon ElastiCache](#)
- [Infrastruktursicherheit in AWS ElastiCache](#)
- [Service-Updates in ElastiCache](#)
- [Häufige Sicherheitslücken und Risiken \(CVE\): Sicherheitslücken, die behoben wurden in ElastiCache](#)

# Datenschutz bei Amazon ElastiCache

Das [Modell der AWS gemeinsamen Verantwortung](#) und geteilter Verantwortung gilt für den Datenschutz in AWS ElastiCache (ElastiCache). Wie in diesem Modell beschrieben, AWS ist es für den Schutz der globalen Infrastruktur verantwortlich, auf der die gesamte AWS Cloud betrieben wird. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieser Inhalt umfasst die Sicherheitskonfiguration und die Verwaltungsaufgaben für die AWS Dienste, die Sie verwenden. Weitere Informationen zum Datenschutz finden Sie im [Abschnitt Datenschutz FAQ](#).

Aus Datenschutzgründen empfehlen wir Ihnen, Ihre AWS Kontoanmeldeinformationen zu schützen und individuelle Konten mit AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto eine Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet TLS, um mit AWS Ressourcen zu kommunizieren.
- Einrichtung API und Protokollierung von Benutzeraktivitäten mit AWS CloudTrail.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen innerhalb der AWS Dienste.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu sichern.

Wir empfehlen dringend, in Freitextfeldern wie z. B. im Feld Name keine sensiblen, identifizierenden Informationen wie Kontonummern von Kunden einzugeben. Dies gilt auch, wenn Sie mit ElastiCache oder anderen AWS Diensten arbeiten, die die Konsole verwenden API, AWS CLI, oder AWS SDKs. Alle Daten, die Sie in Dienste ElastiCache oder andere Dienste eingeben, werden möglicherweise zur Aufnahme in Diagnoseprotokolle aufgenommen. Wenn Sie einem externen Server eine URL zur Verfügung stellen, fügen Sie keine Anmeldeinformationen hinzu, URL um Ihre Anfrage an diesen Server zu überprüfen.

## Themen

- [Datensicherheit bei Amazon ElastiCache](#)

## Datensicherheit bei Amazon ElastiCache

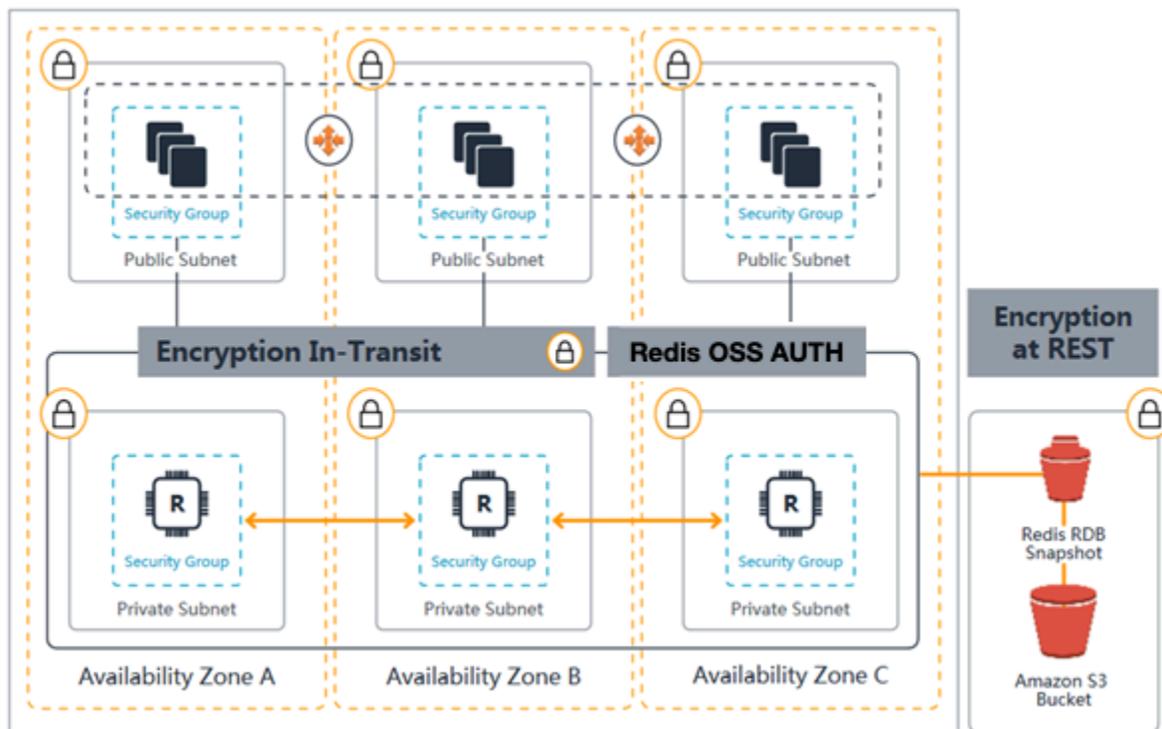
Um die Sicherheit Ihrer Daten zu gewährleisten, bieten Amazon ElastiCache und Amazon Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten auf dem Server.

Amazon ElastiCache (Memcached) bietet Verschlüsselungsfunktionen für Daten in Caches, auf denen Memcached-Versionen 1.6.12 oder höher ausgeführt werden.

Amazon ElastiCache mit Valkey und Redis OSS bietet Verschlüsselungsfunktionen für Daten in Caches, auf denen Valkey 7.2 oder höher und OSS Redis-Versionen 3.2.6 (geplant für, siehe Zeitplan für das [Ende des Lebenszyklus der OSS Redis-Versionen](#)) EOL, 4.0.10 oder höher ausgeführt werden. Amazon unterstützt ElastiCache auch die Authentifizierung von Benutzern mit Valkey und Redis OSS AUTH sowie die Autorisierung von Benutzervorgängen mithilfe von Role-Based Access Control (IAM RBAC).

- Bei der Verschlüsselung während der Übertragung werden Ihre Daten bei der Verschiebung von einem Ort an den anderen, z. B. zwischen Knoten in Ihrem Cluster oder zwischen einem Cache und Ihrer Anwendung, verschlüsselt.
- Bei der Verschlüsselung im Ruhezustand werden Ihre Daten auf dem Laufwerk bei Synchronisierungs- und Sicherungsoperationen verschlüsselt.

ElastiCache unterstützt die Authentifizierung von Benutzern mithilfe der OSS AUTH Befehle Valkey IAM und Redis sowie die Autorisierung von Benutzervorgängen mithilfe der rollenbasierten Zugriffskontrolle (RBAC).



## ElastiCache mit Valkey- und Redis-Sicherheitsdiagramm OSS

### Themen

- [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#)
- [Verschlüsselung im Ruhezustand in ElastiCache](#)
- [Authentifizierung und Autorisierung](#)

## ElastiCache Verschlüsselung bei der Übertragung () TLS

Um Ihre Daten zu schützen, EC2 bieten Amazon ElastiCache und Amazon Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten auf dem Server. Durch die Bereitstellung von Verschlüsselungsfunktionen während der Übertragung ElastiCache erhalten Sie ein Tool, mit dem Sie Ihre Daten schützen können, wenn sie von einem Ort zum anderen übertragen werden.

In allen OSS serverlosen Caches von Valkey oder Redis ist die Verschlüsselung bei der Übertragung aktiviert. Bei selbst entworfenen Clustern können Sie die Verschlüsselung während der Übertragung für eine Replikationsgruppe aktivieren, indem Sie den Parameter auf true (CLI: `--transit-encryption-enabled`) setzen `TransitEncryptionEnabled`, wenn Sie die Replikationsgruppe erstellen. Sie können dies unabhängig davon tun, ob Sie die Replikationsgruppe mit dem AWS Management Console, dem oder dem AWS CLI erstellen. ElastiCache API

Für alle Serverless-Caches ist die Verschlüsselung während der Übertragung aktiviert. Bei selbst entworfenen Clustern können Sie die Verschlüsselung während der Übertragung auf einem Cache-Cluster aktivieren, indem Sie den Parameter `TransitEncryptionEnabled` auf `true` (CLI: `--transit-encryption-enabled`) setzen, wenn Sie den Cache-Cluster mithilfe der Operation `CreateCacheCluster` (CLI: `create-cache-cluster`) erstellen.

## Themen

- [Übersicht über die Verschlüsselung während der Übertragung](#)
- [Verschlüsselungsbedingungen bei der Übertragung \(Valkey und Redis\) OSS](#)
- [Verschlüsselungsbedingungen bei der Übertragung \(Memcached\)](#)
- [Bewährte Methoden für die Verschlüsselung während der Übertragung](#)
- [Weitere Valkey- und Redis-Optionen OSS](#)
- [Aktivierung der Verschlüsselung bei der Übertragung für Memcached](#)
- [Aktivieren von Verschlüsselung während der Übertragung](#)
- [Verbindung zu ElastiCache \(Valkey\) oder Amazon ElastiCache \(RedisOSS\) mit Verschlüsselung während der Übertragung mithilfe von Valkey-CLI herstellen](#)
- [Aktivierung der Verschlüsselung während der Übertragung auf einem selbst entworfenen OSS Redis-Cluster mit Python](#)
- [Bewährte Methoden für die Aktivierung der Verschlüsselung während der Übertragung](#)
- [Mit Openssl \(Memcached\) wird eine Verbindung zu Knoten hergestellt, für die Verschlüsselung während der Übertragung aktiviert ist](#)
- [Einen TLS Memcached-Client mit Java erstellen](#)
- [Einen TLS Memcached-Client erstellen mit PHP](#)

## Übersicht über die Verschlüsselung während der Übertragung

Die ElastiCache Amazon-Verschlüsselung bei der Übertragung ist eine Funktion, mit der Sie die Sicherheit Ihrer Daten an den anfälligsten Stellen erhöhen können — wenn sie von einem Ort zum anderen übertragen werden. Da für die Verschlüsselung und Entschlüsselung der Daten an den Endpunkten eine gewisse Verarbeitung erforderlich ist, kann die Aktivierung der Verschlüsselung während der Übertragung einige Performance-Einbußen zufolge haben. Sie sollten Benchmark-Tests Ihrer Daten mit und ohne Verschlüsselung während der Übertragung durchführen, um die Performance-Einbußen bei Ihren Anwendungsfällen zu bestimmen.

ElastiCache Die Verschlüsselung bei der Übertragung implementiert die folgenden Funktionen:

- Verschlüsselte Clientverbindungen — Client-Verbindungen zu Cache-Knoten sind TLS verschlüsselt.
- Verschlüsselte Serververbindungen – Daten, die zwischen Knoten in einem Cluster übertragen werden, sind verschlüsselt.
- Serverauthentifizierung – Clients können die Verbindung zum richtigen Server authentifizieren.
- Client-Authentifizierung — Mithilfe der Valkey- und OSS AUTH Redis-Funktion kann der Server die Clients authentifizieren.

### Verschlüsselungsbedingungen bei der Übertragung (Valkey und Redis) OSS

Bei der Planung Ihrer selbst entworfenen Cluster-Implementierung sollten Sie die folgenden Einschränkungen für die Verschlüsselung während der ElastiCache Übertragung durch Amazon berücksichtigen:

- Die Verschlüsselung während der Übertragung wird für Replikationsgruppen unterstützt, auf denen Valkey 7.2 und höher sowie die OSS Redis-Versionen 3.2.6, 4.0.10 und höher ausgeführt werden.
- Das Ändern der Verschlüsselungseinstellung bei der Übertragung für einen vorhandenen Cluster wird für Replikationsgruppen unterstützt, auf denen Valkey 7.2 und höher sowie Redis Version 7 und höher ausgeführt wird. OSS
- Die Verschlüsselung während der Übertragung wird nur für Replikationsgruppen unterstützt, die in einem Amazon VPC ausgeführt werden.
- Die Verschlüsselung während der Übertragung wird für Replikationsgruppen, auf denen die folgenden Knotentypen ausgeführt werden, nicht unterstützt: M1, M2.

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Verschlüsselung während der Übertragung wird durch die Festlegung des `TransitEncryptionEnabled`-Parameters auf `true` ausdrücklich aktiviert.
- Stellen Sie sicher, dass Ihr Caching-Client TLS Konnektivität unterstützt und dass Sie sie in der Client-Konfiguration aktiviert haben.
- Die Verwendung der alten Versionen TLS 1.0 und TLS 1.1 ist in allen AWS Regionen für ElastiCache Version 6 und höher veraltet. ElastiCache wird TLS 1.0 und 1.1 bis zum 8. Mai 2025 weiterhin unterstützen. Kunden müssen ihre Client-Software vor diesem Datum aktualisieren.

## Verschlüsselungsbedingungen bei der Übertragung (Memcached)

Bei der Planung Ihrer selbst entworfenen Cluster-Implementierung sollten Sie die folgenden Einschränkungen für die Verschlüsselung während der ElastiCache Übertragung durch Amazon berücksichtigen:

- Verschlüsselung während der Übertragung wird auf Clustern mit der Memcached-Version 1.6.12 und höher unterstützt.
- Die Verschlüsselung bei der Übertragung unterstützt die Versionen 1.2 und 1.3 von Transport Layer Security (TLS).
- Die Verschlüsselung während der Übertragung wird nur für Cluster unterstützt, die in einem Amazon VPC ausgeführt werden.
- Die Verschlüsselung während der Übertragung wird für Replikationsgruppen, auf denen die folgenden Knotentypen ausgeführt werden, nicht unterstützt: M1, M2, M3, R3, T2.

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Verschlüsselung während der Übertragung wird durch die Festlegung des `TransitEncryptionEnabled`-Parameters auf `true` ausdrücklich aktiviert.
- Sie können die Verschlüsselung während der Übertragung nur für einen Cluster aktivieren, wenn Sie den Cluster erstellen. Sie können die Verschlüsselung während der Übertragung nicht aktivieren und deaktivieren, indem Sie einen Cluster ändern.
- Stellen Sie sicher, dass Ihr Caching-Client TLS Konnektivität unterstützt und dass Sie sie in der Client-Konfiguration aktiviert haben.

## Bewährte Methoden für die Verschlüsselung während der Übertragung

- Aufgrund der für die Verschlüsselung und Entschlüsselung der Daten an den Endpunkten erforderliche Verarbeitung, kann die Implementierung der Verschlüsselung während der Übertragung eine Verringerung der Performance zur Folge haben. Führen Sie einen Benchmark-Test der Verschlüsselung während der Übertragung im Vergleich zu keiner Verschlüsselung Ihrer eigenen Daten durch, um deren Auswirkungen auf die Performance Ihrer Implementierung zu ermitteln.
- Da das Erstellen neuer Verbindungen teuer sein kann, können Sie die Auswirkungen der Verschlüsselung bei der Übertragung auf die Leistung verringern, indem Sie Ihre Verbindungen beibehalten. TLS

## Weitere Valkey- und Redis-Optionen OSS

Weitere Informationen zu den für Valkey und Redis verfügbaren Optionen finden Sie unter den OSS folgenden Links.

- [Verschlüsselung im Ruhezustand in ElastiCache](#)
- [Authentifizierung mit den Befehlen Valkey und Redis OSS AUTH](#)
- [Rollenbasierte Zugriffskontrolle \(\) RBAC](#)
- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Identity and Access Management für Amazon ElastiCache](#)

## Aktivierung der Verschlüsselung bei der Übertragung für Memcached

Treffen Sie zum Aktivieren der Verschlüsselung während der Übertragung beim Erstellen eines Memcached-Clusters mit der AWS -Managementkonsole die folgende Auswahl:

- Wählen Sie Memcached als Ihre Engine.
- Wählen Sie die Engine-Version 1.6.12 und höher.
- Wählen Sie unter Encryption in transit (Verschlüsselung während der Übertragung) Enable (Aktivieren) aus.

Informationen zum step-by-step Vorgang finden Sie unter. [Einen Cluster für Valkey oder Redis erstellen OSS](#)

## Aktivieren von Verschlüsselung während der Übertragung

Für alle Serverless-Caches ist die Verschlüsselung während der Übertragung aktiviert. In einem selbst entworfenen Cluster können Sie die Verschlüsselung bei der Übertragung mit dem AWS Management Console, dem oder dem AWS CLI aktivieren. ElastiCache API

## Aktivieren der Verschlüsselung bei der Übertragung mit dem AWS Management Console

Aktivierung der Verschlüsselung bei der Übertragung für einen neuen, selbst entworfenen Cluster mithilfe der AWS Management Console

Wenn Sie einen eigenen Cluster entwerfen, ist bei den Konfigurationen „Dev/Test“ und „Production“ mit der Methode „Easy Create“ die Verschlüsselung während der Übertragung aktiviert. Legen Sie die folgenden Einstellungen fest, wenn Sie die Konfiguration selbst auswählen:

- Wählen Sie die Engine-Version 3.2.6, 4.0.10 und höher.
- Aktivieren Sie für die Option Verschlüsselung während der Übertragung das Kontrollkästchen neben Aktivieren.

Informationen zum step-by-step Prozess finden Sie im Folgenden:

- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Aktivierung der Verschlüsselung während der Übertragung für einen vorhandenen, selbst entworfenen Cluster mithilfe der AWS Management Console

Die Aktivierung der Verschlüsselung während der Übertragung ist ein zweistufiger Vorgang. Sie müssen zuerst den Modus der Übertragungsverchlüsselung auf `preferred` festlegen. In diesem Modus können Ihre Valkey- oder OSS Redis-Clients Verbindungen sowohl über verschlüsselte als auch über unverschlüsselte Verbindungen herstellen. Nachdem Sie alle Ihre Valkey- oder OSS Redis-Clients zur Verwendung verschlüsselter Verbindungen migriert haben, können Sie Ihre Clusterkonfiguration ändern, um den Übertragungsverchlüsselungsmodus auf einzustellen. `required` Wenn Sie den Modus der Verschlüsselung während der Übertragung auf `required` festlegen, werden alle unverschlüsselten Verbindungen gelöscht und nur verschlüsselte Verbindungen werden zugelassen.

Stellen Sie Ihren Transit-Verschlüsselungsmodus auf Preferred ein

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie Valkey-Caches oder OSSRedis-Caches aus den ElastiCache Ressourcen aus, die im Navigationsbereich auf der linken Seite aufgeführt sind.
3. Wählen Sie den Cache aus, den Sie aktualisieren möchten.
4. Wählen Sie das Dropdown-Menü Actions (Aktionen) und dann Modify (Ändern) aus.
5. Wählen Sie im Bereich Sicherheit unter Verschlüsselung während der Übertragung die Option Aktivieren aus.
6. Wählen Sie Preferred (Bevorzugt) als den Transit encryption mode (Modus der Verschlüsselung während der Übertragung) aus.
7. Wählen Sie Preview changes (Änderungsvorschau anzeigen) aus und speichern Sie die Änderungen.

Nachdem Sie alle Ihre Valkey- oder OSS Redis-Clients migriert haben, um verschlüsselte Verbindungen zu verwenden:

Stellen Sie Ihren Transit-Verschlüsselungsmodus auf Erforderlich ein

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie Valkey-Caches oder OSSRedis-Caches aus den ElastiCache Ressourcen aus, die im Navigationsbereich auf der linken Seite aufgeführt sind.
3. Wählen Sie den Cache aus, den Sie aktualisieren möchten.
4. Wählen Sie das Dropdown-Menü Actions (Aktionen) und dann Modify (Ändern) aus.
5. Wählen Sie im Bereich Security (Sicherheit) die Option Required (Erforderlich) als Transit encryption mode (Modus der Verschlüsselung während der Übertragung) aus.
6. Wählen Sie Preview changes (Änderungsvorschau anzeigen) aus und speichern Sie die Änderungen.

Aktivieren Sie die Verschlüsselung bei der Übertragung mit dem AWS CLI

Verwenden Sie den Parameter, um die Verschlüsselung während der Übertragung zu aktivieren, wenn Sie eine Valkey- oder OSS Redis-Replikationsgruppe mithilfe von erstellen. AWS CLI `transit-encryption-enabled`

Aktivierung der Verschlüsselung bei der Übertragung auf einem neuen, selbst entworfenen Cluster für Valkey oder Redis OSS (Clustermodus deaktiviert) () CLI

Verwenden Sie den AWS CLI Vorgang `create-replication-group` und die folgenden Parameter, um eine Valkey- oder OSS Redis-Replikationsgruppe mit Replikaten zu erstellen, für die die Verschlüsselung bei der Übertragung aktiviert ist:

Hauptparameter:

- **--engine**— Muss oder sein. `valkey redis`
- **--engine-version**—Wenn die Engine Redis istOSS, muss es 3.2.6, 4.0.10 oder höher sein.
- **--transit-encryption-enabled**—Erforderlich. Wenn Sie die Verschlüsselung während der Übertragung aktivieren, müssen Sie zudem einen Wert für den `--cache-subnet-group`-Parameter angeben.
- **--num-cache-clusters**—Muss mindestens 1 sein. Der Höchstwert für diesen Parameter ist 6.

Weitere Informationen finden Sie hier:

- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [create-replication-group](#)

Aktivierung der Verschlüsselung während der Übertragung auf einem neuen, selbst entworfenen Cluster für Valkey oder Redis (Clustermodus aktiviert) () OSS CLI

Verwenden Sie den AWS CLI Vorgang `create-replication-group` und die folgenden Parameter, um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) zu erstellen, für die die Verschlüsselung während der Übertragung aktiviert ist:

Hauptparameter:

- **--engine**— Muss oder sein. `valkey redis`
- **--engine-version**—Wenn die Engine Redis istOSS, muss es 3.2.6, 4.0.10 oder höher sein.
- **--transit-encryption-enabled**—Erforderlich. Wenn Sie die Verschlüsselung während der Übertragung aktivieren, müssen Sie zudem einen Wert für den `--cache-subnet-group`-Parameter angeben.
- Verwenden Sie einen der folgenden Parametersätze, um die Konfiguration der Knotengruppen der Replikationsgruppe anzugeben.
  - **--num-node-groups**—Zur Angabe der Anzahl der Shards (Knoten-Gruppen) in dieser Replikationsgruppe. Der Höchstwert für diesen Parameter ist 500.
  - **--replicas-per-node-group**—Definiert die Anzahl der Replikat-Knoten in jeder Knotengruppe. Der hier angegebene Wert wird auf allen Shards in dieser Replikationsgruppe angewendet. Der Höchstwert für diesen Parameter ist 5.
  - **--node-group-configuration**—Spezifiziert die Konfiguration jedes Shards unabhängig.

Weitere Informationen finden Sie hier:

- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [create-replication-group](#)

## Aktivieren der Verschlüsselung während der Übertragung für einen vorhandenen Cluster mithilfe der AWS CLI

Die Aktivierung der Verschlüsselung während der Übertragung ist ein zweistufiger Vorgang. Sie müssen zuerst den Modus der Übertragungsverschlüsselung auf `preferred` festlegen. In diesem Modus können Ihre Valkey- oder OSS Redis-Clients sowohl über verschlüsselte als auch über unverschlüsselte Verbindungen Verbindungen herstellen. Nachdem Sie alle Ihre Valkey- oder OSS Redis-Clients zur Verwendung verschlüsselter Verbindungen migriert haben, können Sie Ihre Clusterkonfiguration ändern, um den Übertragungsverschlüsselungsmodus auf einzustellen. `required` Wenn Sie den Modus der Verschlüsselung während der Übertragung auf `required` festlegen, werden alle unverschlüsselten Verbindungen gelöscht und nur verschlüsselte Verbindungen werden zugelassen.

Verwenden Sie den AWS CLI Vorgang `modify-replication-group` und die folgenden Parameter, um eine Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) zu aktualisieren, für die die Verschlüsselung während der Übertragung deaktiviert ist.

So aktivieren Sie die Verschlüsselung während der Übertragung

1. Stellen Sie unter Verwendung `transit-encryption-mode` der `preferred` folgenden Parameter auf ein
  - **`--transit-encryption-enabled`**—Erforderlich.
  - **`--transit-encryption-mode`** – muss auf `preferred` festgelegt sein.
2. Stellen `transit-encryption-mode` Sie `required` unter Verwendung der folgenden Parameter auf ein:
  - **`--transit-encryption-enabled`**—Erforderlich.
  - **`--transit-encryption-mode`** – muss auf `required` festgelegt sein.

Verbindung zu ElastiCache (Valkey) oder Amazon ElastiCache (RedisOSS) mit Verschlüsselung während der Übertragung mithilfe von Valkey-CLI herstellen

Um auf Daten aus ElastiCache (Redis-OSS) Caches zuzugreifen, für die Verschlüsselung während der Übertragung aktiviert ist, verwenden Sie Clients, die mit Secure Socket Layer (SSL) arbeiten. Sie können `valkey-cli` auch TLS SSL mit/ auf Amazon Linux und Amazon Linux 2 verwenden. Wenn Ihr Client dies nicht unterstützt, können Sie den `stunnel` Befehl auf Ihrem Client-Host verwenden, um einen SSL Tunnel zu den Redis-Knoten zu erstellen. OSS

## Verschlüsselte Verbindung mit Linux

Gehen Sie wie folgt vor, um mit Valkey-CLI eine Verbindung zu einem Valkey- oder OSS Redis-Cluster herzustellen, für den die Verschlüsselung während der Übertragung auf Amazon Linux 2023, Amazon Linux 2 oder Amazon Linux aktiviert ist.

1. Laden Sie das Valkey-CLI-Hilfsprogramm herunter und kompilieren Sie es. Dieses Hilfsprogramm ist in der Valkey-Softwaredistribution enthalten.
2. Geben Sie an der Befehlszeile Ihrer EC2 Instanz die entsprechenden Befehle für die von Ihnen verwendete Linux-Version ein.

### Amazon Linux 2023

Wenn Sie Amazon Linux 2023 verwenden, geben Sie Folgendes ein:

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

Geben Sie dann den folgenden Befehl ein und ersetzen Sie den in diesem Beispiel gezeigten Befehl durch den Endpunkt Ihres Clusters und den Port.

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

Weitere Informationen zum Auffinden des Endpunkts finden Sie unter [Finden Sie Ihre Knotenendpunkte](#).

### Amazon Linux 2

Wenn Sie Amazon Linux 2 verwenden, geben Sie Folgendes ein:

```
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

Wenn Sie Amazon Linux verwenden, geben Sie Folgendes ein:

```
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/
```

Unter Amazon Linux müssen Sie möglicherweise auch die folgenden zusätzlichen Schritte ausführen:

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

3. Nachdem Sie das valkey-cli Utility heruntergeladen und installiert haben, wird empfohlen, den optionalen Befehl auszuführen. `make-test`
4. Um eine Verbindung zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung herzustellen, geben Sie diesen Befehl ein:

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

#### Note

Wenn Sie redis6 auf Amazon Linux 2023 installieren, können Sie jetzt den folgenden Befehl verwenden `redis6-cli: valkey-cli`

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

## Verschlüsselte Verbindung mit Stunnel

Gehen Sie wie folgt vor, um mit Valkey-CLI eine Verbindung zu einem OSS Redis-Cluster herzustellen, für den die Verschlüsselung während der Übertragung mithilfe von Stunnel aktiviert ist.

1. Verwenden Sie SSH, um eine Verbindung zu Ihrem Client herzustellen und zu installieren.  
`stunnel`

```
sudo yum install stunnel
```

2. Führen Sie den folgenden Befehl aus, um eine Datei '/etc/stunnel/valkey-cli.conf' gleichzeitig zu erstellen und zu bearbeiten, um einem oder mehreren Verbindungsparametern einen ElastiCache (Redis-OSS) Cluster-Endpunkt hinzuzufügen. Verwenden Sie dabei die unten angegebene Ausgabe als Vorlage.

```
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 diesem Beispiel hat die Konfigurationsdatei zwei Verbindungen: `valkey-cli` und `valkey-cli-replica`. Dabei werden die Parameter folgendermaßen eingestellt:

- Für `client` wird "yes" eingestellt, um anzugeben, dass diese `stunnel`-Instance ein Client ist.
- `accept` wird die Client-IP zugewiesen. In diesem Beispiel ist der Primärserver auf den OSS Redis-Standard 127.0.0.1 an Port 6379 gesetzt. Das Replikat muss einen anderen Port aufrufen und auf „6380“ festgelegt sein. Sie können die temporären Ports 1024-65535 verwenden. Weitere Informationen finden Sie unter [Ephemeral Ports](#) im VPCAmazon-Benutzerhandbuch.
- `connect` ist auf den OSS Redis-Serverendpunkt eingestellt. Weitere Informationen finden Sie unter [Verbindungsendpunkte finden in ElastiCache](#).

3. Starten `stunnel`.

```
sudo stunnel /etc/stunnel/valkey-cli.conf
```

Verwenden Sie den Befehl `netstat`, um zu bestätigen, dass der Tunnel gestartet wurde.

```
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. Stellen Sie über den lokalen Endpunkt des Tunnels eine Verbindung zum verschlüsselten OSS Redis-Knoten her.

- Wenn bei der Erstellung des ElastiCache (Redis-OSS) Clusters kein AUTH Passwort verwendet wurde, verwendet dieses Beispiel die Valkey-CLI, um eine Verbindung zum ElastiCache (Redis-OSS) Server herzustellen, wobei der vollständige Pfad für Valkey-Cli auf Amazon Linux verwendet wird:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379
```

Wenn AUTH das Passwort bei der Erstellung des OSS Redis-Clusters verwendet wurde, verwendet dieses Beispiel `valkey-cli`, um eine Verbindung zum OSS Redis-Server herzustellen, wobei der vollständige Pfad für `valkey-cli` auf Amazon Linux verwendet wird:

```
/home/ec2-user/redis-7.2.5/src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

ODER

- Ändern Sie das Verzeichnis zu `redis-7.2.5` und gehen Sie wie folgt vor:

Wenn bei der Erstellung des ElastiCache (Redis-OSS) Clusters kein AUTH Passwort verwendet wurde, verwendet dieses Beispiel die Valkey-CLI, um eine Verbindung zum ElastiCache (Redis-OSS) Server herzustellen, wobei der vollständige Pfad für Valkey-Cli auf Amazon Linux verwendet wird:

```
src/valkey-cli -h localhost -p 6379
```

Wenn AUTH das Passwort bei der Erstellung des OSS Redis-Clusters verwendet wurde, verwendet dieses Beispiel `valkey-cli`, um eine Verbindung zum Valkey- oder OSS Redis-Server unter Verwendung des vollständigen Pfads für `valkey-cli` auf Amazon Linux herzustellen:

```
src/valkey-cli -h localhost -p 6379 -a my-secret-password
```

In diesem Beispiel wird Telnet verwendet, um eine Verbindung zum Valkey Redis-Server herzustellen. OSS

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Um die SSL Tunnel zu stoppen und zu schließen, der Stunnel-Prozess `kill`.

```
sudo pkill stunnel
```

### Aktivierung der Verschlüsselung während der Übertragung auf einem selbst entworfenen OSS Redis-Cluster mit Python

Die folgende Anleitung zeigt, wie die Verschlüsselung bei der Übertragung auf einem Redis OSS 7.0-Cluster aktiviert wird, der ursprünglich mit deaktivierter Verschlüsselung während der Übertragung erstellt wurde. TCP und die TLS Clients werden während dieses Vorgangs ohne Ausfallzeiten weiterhin mit dem Cluster kommunizieren.

### Boto3 ruft die benötigten Anmeldeinformationen

(`aws_access_key_id`, `aws_secret_access_key`, und `aws_session_token`) aus den Umgebungsvariablen ab. Diese Anmeldeinformationen werden vorab in dasselbe Bash-Terminal

eingefügt, in dem wir python3 ausführen, um den in diesem Handbuch gezeigten Python-Code zu verarbeiten. Der Code im folgenden Beispiel wurde von einer EC2 Instanz verarbeitet, die in derselben Instanz gestartet wurde VPC, mit der der ElastiCache OSS Redis-Cluster darin erstellt wird.

### Note

- In den folgenden Beispielen wird boto3 SDK für ElastiCache Verwaltungsvorgänge (Cluster- oder Benutzererstellung) und redis-py-cluster redis-py/ für die Datenverarbeitung verwendet.
- Sie müssen mindestens die Boto3-Version (=~) 1.26.39 verwenden, um die Online-Migration mit der Cluster-Modifikation verwenden zu können. TLS API
- ElastiCache unterstützt die TLS Online-Migration nur für Cluster mit Valkey Version 7.2 und höher oder Redis Version 7.0 oder höher. OSS Wenn Sie also einen Cluster haben, auf dem eine OSS Redis-Version vor 7.0 ausgeführt wird, müssen Sie die OSS Redis-Version Ihres Clusters aktualisieren. Weitere Informationen zu Versionsunterschieden finden Sie unter [Wesentliche Unterschiede im Versionsverhalten und in der Kompatibilität mit Redis OSS](#).

## Themen

- [Definieren Sie die Zeichenkettenkonstanten, mit denen der ElastiCache Valkey- oder Redis-Cluster gestartet wird OSS](#)
- [Definieren der Klassen für die Clusterkonfiguration](#)
- [Definieren einer Klasse, die den eigentlichen Cluster darstellt](#)
- [\(Optional\) Erstellen Sie eine Wrapper-Klasse, um die Client-Verbindung zum Valkey- oder Redis-Cluster zu demonstrieren OSS](#)
- [Erstellen der Hauptfunktion, die den Änderungsprozess der Konfiguration der Verschlüsselung während der Übertragung demonstriert](#)

Definieren Sie die Zeichenkettenkonstanten, mit denen der ElastiCache Valkey- oder Redis-Cluster gestartet wird OSS

Definieren wir zunächst einige einfache Python-String-Konstanten, die die Namen der AWS Entitäten enthalten, die zur Erstellung des ElastiCache Clusters erforderlich sind security-group, wieCache

Subnet group, und a. default parameter group All diese AWS Entitäten müssen im Voraus in Ihrem AWS Konto in der Region erstellt werden, die Sie verwenden möchten.

```
#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"
```

## Definieren der Klassen für die Clusterkonfiguration

Lassen Sie uns nun einige einfache Python-Klassen definieren, die eine Konfiguration eines Clusters darstellen und Metadaten über den Cluster wie die Valkey- oder OSS Redis-Version, den Instanztyp und ob die Verschlüsselung bei der Übertragung (TLS) aktiviert oder deaktiviert ist, enthalten.

```
#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

```

## Definieren einer Klasse, die den eigentlichen Cluster darstellt

Lassen Sie uns nun einige einfache Python-Klassen definieren, die den ElastiCache Valkey- oder OSS Redis-Cluster selbst darstellen. Diese Klasse wird über ein Client-Feld verfügen, das einen Boto3-Client für ElastiCache Verwaltungsvorgänge wie das Erstellen des Clusters und das Abfragen des Clusters enthält. 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()
```

(Optional) Erstellen Sie eine Wrapper-Klasse, um die Client-Verbindung zum Valkey- oder Redis-Cluster zu demonstrieren OSS

Jetzt erstellen wir eine Wrapper-Klasse für den `redis-py-cluster`-Client. Diese Wrapper-Klasse unterstützt das Vorbefüllen des Clusters mit einigen Schlüsseln und das anschließende Ausführen zufälliger wiederholter `get`-Befehle.

### Note

Dies ist ein optionaler Schritt, der jedoch den Code der Hauptfunktion vereinfacht, die in einem späteren Schritt erstellt wird.

```
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()
```

Erstellen der Hauptfunktion, die den Änderungsprozess der Konfiguration der Verschlüsselung während der Übertragung demonstriert

Definieren wir nun die Hauptfunktion wie folgt:

1. Erstellen Sie den Cluster mit dem Boto3-Client. ElastiCache
2. Initialisieren Sie den `redis-py-cluster` Client, der sich mit dem Cluster verbindet, mit einer klaren TCP Verbindung ohne. TLS
3. Der `redis-py-cluster`-Client füllt den Cluster vorab mit einigen Daten.
4. Der boto3-Client löst die TLS Migration von „Nein“ zu „Bevorzugt“ aus. TLS TLS
5. Während der Migration auf den Cluster sendet der `redis-py-cluster` TCP Client wiederholte get Operationen an den Cluster `TLSPreferred`, bis die Migration abgeschlossen ist.
6. Nach Abschluss der Migration auf TLS Preferred stellen wir sicher, dass der Cluster die Verschlüsselung bei der Übertragung unterstützt. Anschließend erstellen wir einen `redis-py-cluster` Client, mit TLS dem eine Verbindung zum Cluster hergestellt wird.
7. Wir werden einige get Befehle mit dem neuen TLS Client und dem alten TCP Client senden.
8. Der boto3-Client löst die TLS Migration von „erforderlich“ TLS Preferred nach TLS „erforderlich“ aus.
9. Während der Migration des Clusters auf TLS erforderlich sendet der `redis-py-cluster` TLS Client wiederholte get Operationen an den Cluster, bis die Migration abgeschlossen ist.

```
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)
```

## Bewährte Methoden für die Aktivierung der Verschlüsselung während der Übertragung

Bevor Sie die Verschlüsselung bei der Übertragung aktivieren: Stellen Sie sicher, dass Sie über die richtige Verarbeitung von DNS Datensätzen verfügen

### Note

Während dieses Prozesses ändern und löschen wir alte Endpunkte. Eine falsche Verwendung der Endpunkte kann dazu führen, dass der Valkey- oder OSS Redis-Client alte und gelöschte Endpunkte verwendet, sodass keine Verbindung zum Cluster hergestellt werden kann.

Während der Migration des Clusters von „Nein“ TLS zu „TLSbevorzugt“ werden die alten Einträge pro Knoten beibehalten und die neuen DNS Datensätze pro Knoten werden in einem anderen Format generiert. TLS-aktivierte Cluster verwenden ein anderes Datensatzformat als Cluster. DNS non-TLS-enabled ElastiCache behält beide DNS Datensätze bei, wenn ein Cluster im Verschlüsselungsmodus konfiguriert ist: Vorgezogen, damit Anwendungen und andere Valkey- oder OSS Redis-Clients zwischen ihnen wechseln können. Die folgenden Änderungen in den DNS Datensätzen finden während des TLS -Migrationsprozesses statt:

Beschreibung der Änderungen in den DNS Datensätzen, die bei der Aktivierung der Verschlüsselung während der Übertragung vorgenommen werden

### Für Cluster CME

Wenn ein Cluster auf „Übertragungs-Verschlüsselungsmodus: bevorzugt“ eingestellt ist:

- Die ursprünglichen Cluster-Endpunkte für nicht TLS aktivierte Cluster bleiben aktiv. Es gibt keine Ausfallzeiten, wenn der Cluster vom TLS Verschlüsselungsmodus „ohne“ auf „bevorzugt“ umkonfiguriert wird.

- Neue TLS Valkey- oder OSS Redis-Endpunkte werden generiert, wenn der Cluster auf den Modus `-preferred` gesetzt wird. TLS Diese neuen Endpunkte werden genauso aufgelöst IPs wie die alten (nicht-). TLS
- Der neue TLS Valkey- oder OSS Redis-Konfigurationsendpunkt wird in der ElastiCache Konsole und in der Antwort auf `describe-replication-group` API

Wenn ein Cluster auf „Übertragungs-Verschlüsselungsmodus: erforderlich“ eingestellt ist:

- Alte, nicht TLS aktivierte Endpunkte werden gelöscht. Es wird keine Ausfallzeit von TLS Cluster-Endpunkten geben.
- Sie können ein neues `cluster-configuration-endpoint` von der ElastiCache Konsole oder von der `describe-replication-group` API abrufen.

Für CMD Cluster mit aktiviertem automatischem Failover oder deaktiviertem automatischem Failover

Wenn die Replikationsgruppe auf „Übertragungs-Verschlüsselungsmodus: bevorzugt“ eingestellt ist:

- Der ursprüngliche primäre Endpunkt und der Leser-Endpunkt für den Cluster ohne TLS Aktivierung bleiben aktiv.
- Neue TLS primäre Endpunkte und Leser-Endpunkte werden generiert, wenn der Cluster auf TLS Preferred Modus gesetzt wird. Diese neuen Endpunkte werden auf dieselben IP-Adressen wie die alten (nicht-TLS) aufgelöst.
- Der neue primäre Endpunkt und der Leser-Endpunkt werden in der ElastiCache Konsole und in der Antwort auf die `describe-replication-group` API angezeigt.

Wenn die Replikationsgruppe auf „Übertragungs-Verschlüsselungsmodus: erforderlich“ eingestellt ist:

- Alte Endpunkte (nicht TLS primär) und Lesegeräte werden gelöscht. Es wird keine Ausfallzeit der TLS Cluster-Endpunkte geben.
- Sie können neue Primär- und Leser-Endpunkte von der ElastiCache Konsole oder von der `describe-replication-group` API abrufen.

Die empfohlene Verwendung der Datensätze DNS

Für CME Cluster

- Verwenden Sie den Endpunkt der Cluster-Konfiguration anstelle von DNS Datensätzen pro Knoten im Code Ihrer Anwendung. Die direkte Verwendung von DNS Namen pro Knoten wird nicht empfohlen, da sie sich beim Hinzufügen oder Entfernen von Shards ändern können.
- Kodieren Sie den Endpunkt der Clusterkonfiguration in Ihrer Anwendung nicht fest, da er sich während dieses Vorgangs ändern wird.
- Eine feste Kodierung des Endpunkts der Clusterkonfiguration in Ihrer Anwendung wird nicht empfohlen, da er während dieses Vorgangs geändert werden kann. Wenn die Verschlüsselung während der Übertragung abgeschlossen ist, fragen Sie den Endpunkt der Cluster-Konfiguration mit dem `describe-replication-group` API (wie oben dargestellt (fett gedruckt)) ab und verwenden Sie ab diesem Zeitpunkt die Antwort, die DNS Sie als Antwort erhalten.

#### Für CMD Cluster mit aktiviertem automatischem Failover

- Verwenden Sie den primären Endpunkt und den Leser-Endpunkt anstelle der DNS Namen pro Knoten im Code Ihrer Anwendung, da die alten DNS Namen pro Knoten gelöscht und neue generiert werden, wenn der Cluster von „Nein“ auf „bevorzugt“ migriert wird. TLS TLS Die direkte Verwendung von DNS Namen pro Knoten wird nicht empfohlen, da Sie Ihrem Cluster in future möglicherweise Replikate hinzufügen. Wenn Automatic Failover aktiviert ist, werden außerdem die Rollen des primären Clusters und der Replikate automatisch vom ElastiCache Dienst geändert. Es wird empfohlen, den primären Endpunkt und den Leser-Endpunkt zu verwenden, um den Überblick über diese Änderungen zu behalten. Schließlich hilft Ihnen die Verwendung des Reader-Endpunkts dabei, Ihre Lesevorgänge aus den Replikaten gleichmäßig auf die Replikate im Cluster zu verteilen.
- Es ist eine schlechte Praxis, den primären Endpunkt und den Leser-Endpunkt in Ihrer Anwendung fest zu codieren, da sie während des TLS Migrationsprozesses geändert werden können. Nachdem die Umstellung auf TLS -preferred abgeschlossen ist, fragen Sie den primären Endpunkt und den Reader-Endpunkt mit dem `describe-replication-group` API und verwenden Sie ab diesem Zeitpunkt die Antwort, die DNS Sie als Antwort erhalten. Auf diese Weise können Sie Änderungen an Endpunkten dynamisch verfolgen.

#### Für CMD Cluster mit deaktiviertem automatischen Failover

- Verwenden Sie den primären Endpunkt und den Leser-Endpunkt anstelle der DNS Namen pro Knoten im Code Ihrer Anwendung. Wenn Automatic Failover deaktiviert ist, werden Skalierung, Patching, Failover und andere Verfahren, die automatisch vom ElastiCache Dienst verwaltet werden, wenn Automatic Failover aktiviert ist, stattdessen von Ihnen durchgeführt. Das erleichtert es Ihnen, den Überblick über die verschiedenen Endpunkte zu behalten. Da bei der Migration des

Clusters von „Nein“ auf „TLSBevorzugt“ die alten DNS Namen pro Knoten gelöscht werden und neue generiert werden, sollten Sie die Namen pro Knoten nicht direkt verwenden. TLS DNS Dies ist obligatorisch, damit die Clients während der -Migration eine Verbindung zum Cluster herstellen können. TLS Außerdem profitieren Sie von der gleichmäßigen Verteilung der Lesevorgänge zwischen den Replikaten, wenn Sie den Reader-Endpunkt verwenden, und Sie behalten den Überblick über die DNS -records, wenn Sie Repliken zum Cluster hinzufügen oder daraus löschen.

- Es ist eine schlechte Praxis, den Endpunkt der Clusterkonfiguration in Ihrer Anwendung fest zu codieren, da er während des Migrationsprozesses geändert werden kann. TLS

Bei der Verschlüsselung während der Übertragung: darauf achten, wann der Migrationsprozess abgeschlossen ist

Die Änderung des Verschlüsselungsmodus während der Übertragung erfolgt nicht sofort und kann einige Zeit in Anspruch nehmen. Dies gilt insbesondere für große Cluster. Erst wenn der Cluster die Migration zu TLS -preferred abgeschlossen hat, kann er Verbindungen TCP sowohl TLS akzeptieren als auch bereitstellen. Daher sollten Sie keine Clients erstellen, die versuchen, TLS Verbindungen zum Cluster herzustellen, bis die Verschlüsselung während der Übertragung abgeschlossen ist.

Es gibt mehrere Möglichkeiten, sich benachrichtigen zu lassen, wenn die Verschlüsselung während der Übertragung erfolgreich abgeschlossen wurde oder fehlschlägt (im obigen Codebeispiel nicht dargestellt):

- Verwenden Sie den SNS Dienst, um eine Benachrichtigung zu erhalten, wenn die Verschlüsselung abgeschlossen ist
- Wenn Sie den verwenden `describe-events`API, wird ein Ereignis ausgelöst, wenn die Verschlüsselung abgeschlossen ist
- In der ElastiCache Konsole wird eine Meldung angezeigt, dass die Verschlüsselung abgeschlossen ist

Sie können in Ihrer Anwendung auch Logik implementieren, um festzustellen, ob die Verschlüsselung abgeschlossen ist. Im obigen Beispiel haben wir mehrere Möglichkeiten kennengelernt, wie wir sicherstellen können, dass der Cluster die Migration abschließt:

- Warten, bis der Migrationsprozess beginnt (der Clusterstatus wechselt zu „Wird geändert“) und warten, bis die Änderung abgeschlossen ist (der Clusterstatus wechselt wieder zu „Verfügbar“)
- Es wird bestätigt, dass der Cluster auf `True transit_encryption_enabled` gesetzt wurde, indem die abgefragt wird. `describe-replication-group` API

Nach dem Aktivieren der Verschlüsselung während der Übertragung: sicherstellen, dass die von Ihnen verwendeten Clients ordnungsgemäß konfiguriert sind

Solange sich der Cluster im Modus TLS -preferred befindet, sollte Ihre Anwendung TLS Verbindungen zum Cluster öffnen und nur diese Verbindungen verwenden. Auf diese Weise kommt es bei Ihrer Anwendung nicht zu Ausfallzeiten, wenn die Verschlüsselung während der Übertragung aktiviert wird. Mit dem Befehl `info` im Abschnitt können Sie sicherstellen, dass keine klareren TCP Verbindungen zur Valkey- oder OSS Redis-Engine bestehen. 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
```

Mit Openssl (Memcached) wird eine Verbindung zu Knoten hergestellt, für die Verschlüsselung während der Übertragung aktiviert ist

Um auf Daten von ElastiCache (Memcached-) Knoten zuzugreifen, für die Verschlüsselung bei der Übertragung aktiviert ist, müssen Sie Clients verwenden, die mit Secure Socket Layer (SSL) arbeiten. Sie können Openssl `s_client` auch auf Amazon Linux und Amazon Linux 2 verwenden.

Verwenden Sie Openssl `s_client`, um sich mit einem Memcached-Cluster zu verbinden, der mit Verschlüsselung während der Übertragung auf Amazon Linux 2 oder Amazon Linux aktiviert ist, wie folgt:

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

### Einen TLS Memcached-Client mit Java erstellen

Um einen Client im TLS Modus zu erstellen, gehen Sie wie folgt vor, um den Client mit dem entsprechenden zu initialisieren: `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");
 }
}
```

## Einen TLS Memcached-Client erstellen mit PHP

Um einen Client im TLS Modus zu erstellen, gehen Sie wie folgt vor, um den Client mit dem entsprechenden zu initialisieren: 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);
}
```

Weitere Informationen zur Verwendung des PHP Clients finden Sie unter [Installieren des ElastiCache-Cluster-Clients-für-PHP](#).

## Verschlüsselung im Ruhezustand in ElastiCache

Um Ihre Daten zu schützen, bieten Amazon ElastiCache und Amazon S3 verschiedene Möglichkeiten, den Zugriff auf Daten in Ihrem Cache einzuschränken. Weitere Informationen erhalten Sie unter [Amazon VPCs und ElastiCache Sicherheit](#) und [Identity and Access Management für Amazon ElastiCache](#).

ElastiCache Die Verschlüsselung im Ruhezustand ist eine Funktion zur Erhöhung der Datensicherheit durch Verschlüsselung von Daten auf der Festplatte. In einem Serverless-Cache ist sie immer aktiviert. Wenn die Funktion aktiviert ist, werden folgenden Aspekte verschlüsselt:

- Festplatte bei Synchronisierung, Backup und Austausch
- In Amazon S3 gespeicherte Backups

Daten, die auf SSDs (Solid-State-Laufwerken) in Clustern mit aktivierter Datenklassifizierung gespeichert sind, sind immer verschlüsselt.

ElastiCache bietet standardmäßige (vom Service verwaltete) Verschlüsselung im Ruhezustand sowie die Möglichkeit, Ihre eigenen symmetrischen, vom Kunden verwalteten AWS KMS Schlüssel im [AWS Key Management Service](#) () zu verwenden. KMS Wählen Sie bei der Sicherung des Caches unter Verschlüsselungsoptionen aus, ob Sie den Standardverschlüsselungsschlüssel oder einen vom Kunden verwalteten Schlüssel verwenden möchten. Weitere Informationen finden Sie unter [Aktivieren der Verschlüsselung im Ruhezustand](#).

### Note

Die Standardverschlüsselung (vom Service verwaltet) ist die einzige Option, die in den Regionen GovCloud (USA) verfügbar ist.

### Important

Wenn Sie die Verschlüsselung im Ruhezustand auf einem vorhandenen, selbst entworfenen Valkey- oder OSS Redis-Cluster aktivieren, müssen Sie Ihre bestehende Replikationsgruppe löschen, nachdem Sie die Sicherung und Wiederherstellung für die Replikationsgruppe ausgeführt haben.

Die Verschlüsselung im Ruhezustand kann nur für einen Cache aktiviert werden, wenn dieser erstellt wird. Da zum Verschlüsseln und Entschlüsseln der Daten ein gewisses Maß an Verarbeitung erforderlich ist, kann die Aktivierung der Verschlüsselung der Daten im Ruhezustand bei diesen Vorgängen zu Leistungseinbußen führen. Sie sollten Ihre Daten mit und ohne Verschlüsselung im Ruhezustand vergleichen, um die Auswirkungen auf die Leistung für Ihre Anwendungsfälle zu ermitteln.

## Themen

- [Bedingungen für die Verschlüsselung im Ruhezustand](#)
- [Verwenden Sie vom Kunden verwaltete Schlüssel von AWS KMS](#)
- [Aktivieren der Verschlüsselung im Ruhezustand](#)
- [Weitere Informationen finden Sie unter:](#)

## Bedingungen für die Verschlüsselung im Ruhezustand

Bei der Planung der Implementierung ElastiCache der Verschlüsselung im Ruhezustand sollten Sie die folgenden Einschränkungen bei der ElastiCache Verschlüsselung im Ruhezustand berücksichtigen:

- Die Verschlüsselung im Ruhezustand wird für Replikationsgruppen unterstützt, auf denen Valkey 7.2 und höher ausgeführt wird, sowie auf OSS Redis-Versionen (3.2.6 geplant, siehe Zeitplan für das [Ende des Lebenszyklus der OSS Redis-Versionen](#)) EOL, 4.0.10 oder höher.
- Die Verschlüsselung im Ruhezustand wird nur für Replikationsgruppen unterstützt, die in einem Amazon VPC ausgeführt werden.
- Die Verschlüsselung im Ruhezustand wird nur für Replikationsgruppen unterstützt, die die folgenden Knotentypen ausführen.
  - R6gd, R6g, R5, R4, R3
  - M6g, M5, M4, M3
  - T4g, T3, T2

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Die Verschlüsselung im Ruhezustand wird durch die explizite Festlegung des `AtRestEncryptionEnabled`-Parameters auf `true` aktiviert.
- Sie können die Verschlüsselung im Ruhezustand nur für eine Replikationsgruppe aktivieren, wenn Sie die Replikationsgruppe erstellen. Sie können die Verschlüsselung während der Übertragung

nicht aktivieren und deaktivieren, indem Sie eine Replikationsgruppe ändern. Informationen zur Implementierung der Verschlüsselung im Ruhezustand bei vorhandenen Replikationsgruppen finden Sie unter [Aktivieren der Verschlüsselung im Ruhezustand](#).

- Wenn ein Cluster einen Knotentyp aus der R6GD-Familie verwendet, werden die darauf SSD gespeicherten Daten verschlüsselt, unabhängig davon, ob die Verschlüsselung im Ruhezustand aktiviert ist oder nicht.
- Die Option, vom Kunden verwaltete Schlüssel für die Verschlüsselung im Ruhezustand zu verwenden, ist in den Regionen AWS GovCloud (us-gov-east-1 und us-gov-west-1) nicht verfügbar.
- Wenn ein Cluster einen Knotentyp aus der R6GD-Familie verwendet, werden die darauf SSD gespeicherten Daten mit dem ausgewählten vom Kunden verwalteten AWS KMS Schlüssel (oder der vom Service verwalteten Verschlüsselung in AWS GovCloud Regionen) verschlüsselt.
- Mit Memcached wird die Verschlüsselung im Ruhezustand nur in serverlosen Caches unterstützt.
- Bei Verwendung von Memcached ist die Option, vom Kunden verwaltete Schlüssel für die Verschlüsselung im Ruhezustand zu verwenden, in AWS GovCloud den Regionen (-1 und -1) nicht verfügbar. us-gov-east us-gov-west

Durch die Implementierung der Verschlüsselung der Daten im Ruhezustand kann die Leistung während der Sicherungs- und Knoten-Synchronisierungsvorgänge reduziert sein. Vergleichen Sie die Verschlüsselung der Daten im Ruhezustand mit Ihren eigenen unverschlüsselten Daten, um die Auswirkungen auf die Leistung Ihrer Implementierung zu ermitteln.

Verwenden Sie vom Kunden verwaltete Schlüssel von AWS KMS

ElastiCache unterstützt symmetrische, vom Kunden verwaltete AWS KMS Schlüssel (KMSSchlüssel) für die Verschlüsselung im Ruhezustand. Kundenverwaltete KMS Schlüssel sind Verschlüsselungsschlüssel, die Sie in Ihrem AWS Konto erstellen, besitzen und verwalten. Weitere Informationen finden Sie unter [AWS KMSSchlüssel](#) im AWS Key Management Service Developer Guide. Die Schlüssel müssen erstellt werden, AWS KMS bevor sie mit verwendet werden können ElastiCache.

Informationen zum Erstellen von AWS KMS Stammschlüsseln finden Sie unter [Creating Keys](#) im AWS Key Management Service Developer Guide.

ElastiCache ermöglicht Ihnen die Integration mit AWS KMS. Weitere Informationen finden Sie unter [Verwendung von Berechtigungen](#) im Entwicklerhandbuch zum AWS -Schlüsselverwaltungsdienst.

Es sind keine Kundenaktionen erforderlich, um die ElastiCache Amazon-Integration mit zu aktivieren AWS KMS.

Der `kms:ViaService` Bedingungsschlüssel beschränkt die Verwendung eines AWS KMS Schlüssels (KMSSchlüssels) auf Anfragen von bestimmten AWS Diensten. Um `kms:ViaService` mit zu verwenden ElastiCache, schließen Sie beide `ViaService` Namen in den Wert des Bedingungsschlüssels ein: `elasticache.AWS_region.amazonaws.com` und `unddax.AWS_region.amazonaws.com`. Weitere Informationen finden Sie unter [kms: ViaService](#).

Sie können [AWS CloudTrail](#) damit die Anfragen verfolgen, an die Amazon in AWS Key Management Service Ihrem Namen ElastiCache sendet. Alle API Aufrufe, die sich auf vom Kunden verwaltete Schlüssel AWS Key Management Service beziehen, haben entsprechende CloudTrail Protokolle. Wenn Sie den [ListGrants](#) KMSAPI Aufruf aufrufen, können Sie sich auch die Zuschüsse ansehen, die ElastiCache dadurch entstehen.

Ist eine Replikationsgruppe mit einem vom Kunden verwalteten Schlüssel verschlüsselt, werden alle Backups für die Replikationsgruppe wie folgt verschlüsselt:

- Tägliche automatische Backups werden mit dem vom Kunden verwalteten Schlüssel verschlüsselt, der dem Cluster zugeordnet ist.
- Die letzte Sicherung, die beim Löschen der Replikationsgruppe erstellt wird, wird ebenfalls mit dem kundenverwalteten Schlüssel verschlüsselt, der der Replikationsgruppe zugeordnet ist.
- Manuell erstellte Backups werden standardmäßig so verschlüsselt, dass sie den KMS Schlüssel verwenden, der der Replikationsgruppe zugeordnet ist. Mit einem anderen kundenverwalteten Schlüssel können Sie dies außer Kraft setzen.
- Als Standardeinstellung wird beim Kopieren einer Sicherung ein vom Kunden verwalteter Schlüssel verwendet, der mit der Quellsicherung verknüpft ist. Mit einem anderen kundenverwalteten Schlüssel können Sie dies außer Kraft setzen.

#### Note

- Kundenverwaltete Schlüssel können beim Exportieren von Sicherungen in den von Ihnen ausgewählten Amazon S3-Bucket nicht verwendet werden. Alle in Amazon S3 exportierten Sicherungen werden jedoch mit [serverseitiger Verschlüsselung](#) verschlüsselt. Sie können wählen, ob Sie die Sicherungsdatei in ein neues S3-Objekt kopieren und mit einem vom Kunden verwalteten KMS Schlüssel verschlüsseln, die Datei in einen anderen S3-Bucket

kopieren, der mit einer Standardverschlüsselung mithilfe eines KMS Schlüssels eingerichtet ist, oder eine Verschlüsselungsoption in der Datei selbst ändern.

- Außerdem können Sie vom Kunden verwaltete Schlüssel verwenden, um manuell erstellte Backups für Replikationsgruppen zu verschlüsseln, die keine vom Kunden verwalteten Schlüssel für die Verschlüsselung verwenden. Mit dieser Option wird die in Amazon S3 gespeicherte Sicherungsdatei mit einem KMS Schlüssel verschlüsselt, obwohl die Daten in der ursprünglichen Replikationsgruppe nicht verschlüsselt sind.

Bei der Wiederherstellung von einer Sicherung können Sie aus verschiedenen Verschlüsselungsoptionen wählen, ähnlich den verfügbaren Verschlüsselungsoptionen beim Erstellen einer neuen Replikationsgruppe.

- Wenn Sie den Schlüssel löschen oder [deaktivieren](#) und für den Schlüssel, den Sie zur Verschlüsselung eines Caches verwendet haben, [Genehmigungen zurückziehen](#), kann der Cache nicht wiederhergestellt werden. Mit anderen Worten, sie kann nach einem Hardwarefehler nicht geändert oder wiederhergestellt werden. AWS KMS löscht Root-Schlüssel erst nach einer Wartezeit von mindestens sieben Tagen. Nach dem Löschen des Schlüssels können Sie einen anderen vom Kunden verwalteten Schlüssel verwenden, um eine Sicherungskopie für Archivierungszwecke zu erstellen.
- Bei der automatischen Schlüsselrotation bleiben die Eigenschaften Ihrer AWS KMS Stammschlüssel erhalten, sodass die Rotation keine Auswirkungen auf Ihre Fähigkeit hat, auf Ihre Daten zuzugreifen. ElastiCache Verschlüsselte ElastiCache Amazon-Caches unterstützen keine manuelle Schlüsselrotation, bei der ein neuer Root-Schlüssel erstellt und alle Verweise auf den alten Schlüssel aktualisiert werden. Weitere Informationen finden Sie unter [Rotation von AWS KMS Schlüsseln](#) im AWS Key Management Service Developer Guide.
- Für die Verschlüsselung eines ElastiCache Caches mithilfe eines KMS Schlüssels ist ein Grant pro Cache erforderlich. Diese Genehmigung gilt für die Lebensdauer des Caches. Zusätzlich wird eine Genehmigung pro Backup bei der Backup-Erstellung verwendet. Diese Genehmigung wird zurückgezogen, sobald das Backup erstellt wurde.
- Weitere Informationen zu AWS KMS Zuschüssen und Beschränkungen finden Sie unter [Grenzwerte](#) im AWS Key Management Service Developer Guide.

## Aktivieren der Verschlüsselung im Ruhezustand

Für alle Serverless-Caches ist die Verschlüsselung im Ruhezustand aktiviert.

Wenn Sie einen selbst entworfenen Cluster erstellen, können Sie die Verschlüsselung im Ruhezustand aktivieren, indem Sie den Parameter `AtRestEncryptionEnabled` auf `true` festlegen. Sie können die Verschlüsselung der Daten im Ruhezustand für vorhandene Replikationsgruppen nicht aktivieren.

Sie können die Verschlüsselung im Ruhezustand aktivieren, wenn Sie einen ElastiCache Cache erstellen. Sie können dies mit dem AWS Management Console, der AWS CLI, dem oder dem ElastiCache API tun.

Beim Erstellen eines Caches können Sie eine der folgenden Optionen auswählen:

- Standard – Diese Option verwendet serviceverwaltete Verschlüsselung im Ruhezustand.
- Vom Kunden verwalteter Schlüssel — Mit dieser Option können Sie die Schlüssel-ID/ ARN von AWS KMS für die Verschlüsselung im Ruhezustand angeben.

Informationen zum Erstellen von AWS KMS Root-Schlüsseln finden Sie unter [Create Keys](#) im AWS Key Management Service Developer Guide

## Inhalt

- [Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS Management Console](#)
- [Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS CLI](#)

Aktivierung der Verschlüsselung im Ruhezustand auf einem vorhandenen, selbst entworfenen Valkey- oder Redis-Cluster OSS

Sie können die Verschlüsselung im Ruhezustand nur aktivieren, wenn Sie eine Valkey- oder Redis-Replikationsgruppe erstellen. Gehen Sie folgendermaßen vor, wenn Sie über eine vorhandene Replikationsgruppe verfügen, für die Sie die Verschlüsselung der Daten im Ruhezustand aktivieren möchten.

Aktivieren der Verschlüsselung der Daten im Ruhezustand für eine vorhandene Replikationsgruppe

1. Erstellen Sie eine manuelle Sicherung Ihrer vorhandenen Replikationsgruppe. Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).
2. Erstellen Sie eine neue Replikationsgruppe, indem Sie sie aus der Sicherung wiederherstellen. Aktivieren Sie in der neuen Replikationsgruppe die Verschlüsselung der Daten im Ruhezustand. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).

3. Aktualisieren Sie die Endpunkte in Ihrer Anwendung so, dass sie auf die neue Replikationsgruppe verweisen.
4. Löschen Sie die alte Replikationsgruppe. Weitere Informationen finden Sie unter [Löschen eines Clusters in ElastiCache](#) oder [Löschen einer Replikationsgruppe](#).

Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS Management Console

Aktivieren der Verschlüsselung im Ruhezustand für einen Serverless-Cache (Konsole)

Für alle Serverless-Caches ist die Verschlüsselung im Ruhezustand aktiviert. Standardmäßig wird ein AWS-eigener KMS Schlüssel zum Verschlüsseln von Daten verwendet. Um Ihren eigenen AWS KMS Schlüssel zu wählen, treffen Sie die folgenden Auswahlen:

- Erweitern Sie den Abschnitt Standardeinstellungen.
- Wählen Sie im Abschnitt Standardeinstellungen die Option Standardeinstellungen anpassen aus.
- Wählen Sie im Abschnitt Sicherheit die Option Sicherheitseinstellungen anpassen aus.
- Wählen Sie CMK unter Einstellung des Verschlüsselungsschlüssels die Option Vom Kunden verwaltet aus.
- Wählen Sie unter der Einstellung AWS KMS -Schlüssel einen Schlüssel aus.

Aktivieren der Verschlüsselung im Ruhezustand für einen selbst entworfenen Cluster (Konsole)

Wenn Sie einen eigenen Cache entwerfen, ist bei den Konfigurationen „Dev/Test“ und „Production“ mit der Methode „Easy Create“ die Verschlüsselung im Ruhezustand mithilfe des Schlüssels Standard aktiviert. Legen Sie die folgenden Einstellungen fest, wenn Sie die Konfiguration selbst auswählen:

- Wählen Sie Version 3.2.6, 4.0.10 oder höher als Ihre Engine-Version.
- Aktivieren Sie für die Option Verschlüsselung im Ruhezustand das Kontrollkästchen neben Aktivieren.
- Wählen Sie entweder einen Standardschlüssel oder einen vom Kunden verwalteten Schlüssel.  
CMK

Das step-by-step Verfahren finden Sie im Folgenden:

- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

- [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS CLI

Um die Verschlüsselung im Ruhezustand zu aktivieren, wenn Sie einen Valkey- oder OSS Redis-Cluster mithilfe von erstellen AWS CLI, verwenden Sie beim Erstellen einer Replikationsgruppe den `at-rest-encryption-enabled` Parameter --.

Aktivierung der Verschlüsselung im Ruhezustand auf einem Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) () CLI

Der folgende Vorgang erstellt die Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) `my-classic-rg` mit drei Knoten (-- `num-cache-clusters`), einem primären und zwei Read Replicas. Die Verschlüsselung im Ruhezustand ist für diese Replikationsgruppe aktiviert (--). `at-rest-encryption-enabled`

Die folgenden Parameter und ihre Werte sind erforderlich, um die Verschlüsselung für diese Replikationsgruppe zu aktivieren:

Hauptparameter

- **--engine**— Muss oder sein. `valkey redis`
- **--engine-version**—Wenn die Engine Redis istOSS, muss es 3.2.6, 4.0.10 oder höher sein.
- **--at-rest-encryption-enabled**—Erforderlich, um die Verschlüsselung von Daten im Ruhezustand zu aktivieren.

Example 1: Valkey- oder Redis-Cluster OSS (Clustermodus deaktiviert) mit Replikaten

Für Linux, macOS oder Unix:

```
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
```

Für 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 ^
```

Weitere Informationen finden Sie unter:

- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) von Grund auf neu erstellen \(\)AWS CLI](#)
- [create-replication-group](#)

Aktivierung der Verschlüsselung im Ruhezustand auf einem Cluster für Valkey oder Redis OSS (Clustermodus aktiviert) () CLI

Der folgende Vorgang erstellt die Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) *my-clustered-rg* mit drei Knotengruppen oder Shards (--). num-node-groups Jeder hat drei Knoten, einen primären und zwei Read Replicas (--). replicas-per-node-group Die Verschlüsselung im Ruhezustand ist für diese Replikationsgruppe aktiviert (-- at-rest-encryption-enabled).

Die folgenden Parameter und ihre Werte sind erforderlich, um die Verschlüsselung für diese Replikationsgruppe zu aktivieren:

Hauptparameter

- **--engine**— Muss *valkey* oder *redis* sein.
- **--engine-version**—Wenn die Engine *Redis* ist, muss es *4.0.10* oder höher sein.
- **--at-rest-encryption-enabled**—Erforderlich, um die Verschlüsselung von Daten im Ruhezustand zu aktivieren.
- **--cache-parameter-group**—Muss *default-redis4.0.cluster.on* oder eine davon abgeleitete sein, um diese zu einer Cluster-Modus-fähigen Replikationsgruppe zu machen.

## Example 2: Ein Valkey- oder OSS Redis-Cluster (Clustermodus aktiviert)

Für Linux, macOS oder 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
```

Für 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
```

Weitere Informationen finden Sie unter:

- [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus aktiviert\) von Grund auf neu erstellen \(AWS CLI\)](#)
- [create-replication-group](#)

Weitere Informationen finden Sie unter:

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Identity and Access Management für Amazon ElastiCache](#)

## Authentifizierung und Autorisierung

AWS Identity and Access Management (IAM) ist ein Webdienst, mit dem Sie den Zugriff auf AWS Ressourcen sicher kontrollieren können. ElastiCache unterstützt die Authentifizierung von Benutzern mithilfe IAM der OSS AUTH Befehle Valkey und Redis sowie die Autorisierung von Benutzervorgängen mithilfe der rollenbasierten Zugriffssteuerung (). RBAC

### Themen

- [Rollenbasierte Zugriffskontrolle \(\) RBAC](#)
- [Authentifizierung mit den Befehlen Valkey und Redis OSS AUTH](#)
- [Deaktivieren der Zugriffskontrolle für einen ElastiCache Valkey- oder Redis-Cache OSS](#)

### Rollenbasierte Zugriffskontrolle () RBAC

Anstatt Benutzer mit dem Valkey- und OSS AUTH Redis-Befehl zu authentifizieren, wie unter beschrieben [Authentifizierung mit den Befehlen Valkey und Redis OSS AUTH](#), können Sie in Valkey 7.2 und höher und ab Redis OSS 6.0 eine Funktion namens Role-Based Access Control () verwenden. RBAC RBACist auch die einzige Möglichkeit, den Zugriff auf serverlose Caches zu kontrollieren.

Im Gegensatz zu Valkey und Redis OSSAUTH, bei denen alle authentifizierten Clients vollen Cache-Zugriff haben, wenn ihr Token authentifiziert ist, RBAC können Sie den Cache-Zugriff über Benutzergruppen steuern. Diese Benutzergruppen dienen dazu, den Zugriff auf Caches zu organisieren.

Mit RBAC erstellen Sie Benutzer und weisen ihnen mithilfe einer Zugriffszeichenfolge bestimmte Berechtigungen zu, wie im Folgenden beschrieben. Sie weisen die Benutzer Benutzergruppen zu, denen eine bestimmte Rolle zugewiesen ist (Administratoren, Personalabteilung), die dann in einem oder mehreren ElastiCache Caches bereitgestellt werden. Auf diese Weise können Sie Sicherheitsgrenzen zwischen Clients einrichten, die denselben Valkey- oder OSS Redis-Cache oder dieselben Caches verwenden, und verhindern, dass Clients gegenseitig auf die Daten zugreifen.

#### Note

Bei der Verwendung RBAC mit Valkey-Clustern müssen Sie Benutzern und Benutzergruppen weiterhin die Engine „redis“ zuweisen.

RBAC wurde entwickelt, um die Einführung von [ACL](#) in OSS Redis 6 zu unterstützen. Bei der Verwendung RBAC mit Ihrem ElastiCache Valkey- oder OSS Redis-Cache gibt es einige Einschränkungen:

- Sie können keine Passwörter in einer Zugriffsfolge angeben. Sie legen Passwörter mit [CreateUser](#) oder [ModifyUser](#) Anrufen fest.
- Für Benutzerrechte übergeben Sie `on` und `off` als Teil der Zugriffszeichenfolge. Wenn in der Zugriffszeichenfolge keine angegeben ist, wird dem Benutzer `off` zugewiesen und er hat keine Zugriffsrechte für den Cache.
- Sie können keine verbotenen oder umbenannten Befehle verwenden. Gibt man einen verbotenen oder einen umbenannten Befehl an, wird eine Ausnahme ausgelöst. Wenn Sie Zugriffskontrolllisten (ACLs) für einen umbenannten Befehl verwenden möchten, geben Sie den ursprünglichen Namen des Befehls an, d. h. den Namen des Befehls vor der Umbenennung.
- Sie können den `reset`-Befehl als Teil einer Zugriffszeichenfolge nicht benutzen. Sie geben Passwörter mit API Parametern an und ElastiCache (RedisOSS) verwaltet Passwörter. Daher können Sie `reset` nicht nutzen, da es alle Kennwörter für einen Benutzer entfernen würde.
- Redis OSS 6 führt den [ACLLIST](#) Befehl ein. Dieser Befehl gibt eine Liste von Benutzern zusammen mit den ACL Regeln zurück, die für jeden Benutzer gelten. ElastiCache (RedisOSS) unterstützt den `ACL LIST` Befehl, bietet jedoch keine Unterstützung für Passwort-Hashes wie OSS Redis. Mit ElastiCache (RedisOSS) können Sie die Operation [describe-users](#) verwenden, um ähnliche Informationen abzurufen, einschließlich der Regeln, die in der Zugriffszeichenfolge enthalten sind. Jedoch ruft [describe-user](#) kein Benutzerkennwort ab.

Andere schreibgeschützte Befehle, die von ElastiCache with Valkey und Redis unterstützt werden, sind, und. OSS [ACLWHOAMIACLUSERSACL](#) ElastiCache with Valkey und Redis unterstützt OSS keine anderen schreibbasierten Befehle. ACL

- Die folgenden Einschränkungen gelten:

Ressource	Maximum erlaubt
Benutzer pro Benutzergruppe	100
Anzahl Benutzer	1000
Anzahl der Benutzergruppen	100

Die Verwendung RBAC mit ElastiCache (RedisOSS) wird im Folgenden ausführlicher beschrieben.

## Themen

- [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#)
- [ElastiCache Mit Valkey oder Redis auf einen Cache beantragen RBAC OSS](#)
- [Migration von zu AUTH RBAC](#)
- [Migration von nach RBAC AUTH](#)
- [Automatisches Rotieren von Passwörtern für Benutzer](#)
- [Authentifizierung mit IAM](#)

## Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge

Um Berechtigungen für einen ElastiCache (Redis-OSS) Cache anzugeben, erstellen Sie eine Zugriffszeichenfolge und weisen sie einem Benutzer zu, indem Sie entweder das AWS CLI oder verwenden. AWS Management Console

Zugriffszeichenfolgen werden als eine Liste von durch Leerzeichen getrennten Regeln definiert, die für den Benutzer angewendet werden. Sie definieren, welche Befehle ein Benutzer ausführen kann und welche Schlüssel ein Benutzer benutzen kann. Um einen Befehl auszuführen, muss ein Benutzer Zugriff auf den ausgeführten Befehl und alle Schlüssel haben, auf die mit dem Befehl zugegriffen wird. Regeln werden von links nach rechts kumulativ angewendet und anstelle der angegebenen Zeichenfolge kann eine einfachere Zeichenfolge verwendet werden, wenn Redundanzen in der angegebenen Zeichenfolge vorhanden sind.

Hinweise zur Syntax der ACL Regeln finden Sie unter [ACL](#).

Im folgenden Beispiel wird durch die Zugriffszeichenfolge ein aktiver Benutzer dargestellt, der Zugriff auf alle verfügbaren Schlüssel und Befehle hat.

```
on ~* +@all
```

Die Syntax der Zugriffszeichenfolge wird wie folgt verteilt:

- `on` – Der Benutzer ist ein aktiver Benutzer.
- `~*` – Zugriff auf alle verfügbaren Schlüssel ist erlaubt.
- `+@all` – Zugriff auf alle verfügbaren Befehle ist erlaubt.

Die vorgenannten Einstellungen sind am wenigsten restriktiv. Sie können diese Einstellungen ändern und sie sicherer zu machen.

Das folgende Beispiel zeigt einen Benutzer, der nur Lesezugriff auf Schlüssel hat, die mit dem Schlüsselbereich "app:." beginnen

```
on ~app:.* -@all +@read
```

Sie können diese Berechtigungen weiter verfeinern, indem Sie die Befehle auflisten, auf die der Benutzer zugreifen kann:

+*command1*— Der Zugriff des Benutzers auf Befehle ist beschränkt auf *Befehl1*.

+@category – Der Zugriff des Benutzers auf Befehle ist auf eine Kategorie von Befehlen beschränkt.

Informationen zum Zuweisen einer Zugriffszeichenfolge zu einem Benutzer finden Sie unter [Benutzer und Benutzergruppen mit der Konsole erstellen und CLI](#).

Wenn Sie einen vorhandenen Workload zu migrieren ElastiCache, können Sie die Zugriffszeichenfolge durch einen Aufruf abrufen `ACL LIST`, wobei der Benutzer und alle Kennwort-Hashes ausgenommen sind.

Für Redis OSS Version 6.2 und höher wird auch die folgende Syntax für die Zugriffszeichenfolge unterstützt:

- &\* – Zugriff auf alle verfügbaren Kanäle ist erlaubt.

Für Redis OSS Version 7.0 und höher wird auch die folgende Syntax für Zugriffszeichenfolgen unterstützt:

- | – kann zum Blockieren von Unterbefehlen verwendet werden (z. B. „-config|set“).
- %R~<pattern> – fügt das angegebene Leseschlüsselmuster hinzu. Dies verhält sich ähnlich wie das reguläre Schlüsselmuster, gewährt jedoch nur die Erlaubnis, aus Schlüsseln zu lesen, die dem angegebenen Muster entsprechen. Weitere Informationen finden Sie unter [Wichtigen Berechtigungen](#).
- %W~<pattern> – fügt das angegebene Schreibschlüsselmuster hinzu. Dies verhält sich ähnlich wie das reguläre Schlüsselmuster, gewährt jedoch nur die Erlaubnis, in Schlüssel zu schreiben, die dem angegebenen Muster entsprechen. Weitere Informationen finden Sie unter [ACL Schlüsselberechtigungen](#).
- %RW~<pattern> – Alia für ~<pattern>.

- (`<rule list>`) – erstellt einen neuen Selektor, mit dem die Regeln verglichen werden sollen. Selektoren werden nach den Benutzerberechtigungen und in der Reihenfolge bewertet, in der sie definiert sind. Wenn ein Befehl entweder den Benutzerberechtigungen oder einem beliebigen Selektor entspricht, ist er zulässig. Weitere Informationen finden Sie unter [ACLSelektoren](#).
- `clearselectors` – löscht alle an den Benutzer angehängten Selektoren.

## ElastiCache Mit Valkey oder Redis auf einen Cache beantragen RBAC OSS

Für die Verwendung ElastiCache mit Valkey oder Redis OSS RBAC gehen Sie wie folgt vor:

1. Erstellung eines oder mehrerer Benutzer.
2. Erstellen Sie eine Benutzergruppe und fügen Sie der Gruppe Benutzer hinzu.
3. Weisen Sie die Benutzergruppe einen Cache zu, bei dem die Verschlüsselung während der Übertragung aktiviert ist.

Diese Schritte werden im Folgenden detailliert beschrieben.

### Themen

- [Benutzer und Benutzergruppen mit der Konsole erstellen und CLI](#)
- [Benutzergruppen mit der Konsole verwalten und CLI](#)
- [Zuweisen von Benutzergruppen zu Serverless-Caches](#)
- [Zuweisung von Benutzergruppen zu Replikationsgruppen](#)

## Benutzer und Benutzergruppen mit der Konsole erstellen und CLI

Die Benutzerinformationen für RBAC Benutzer sind eine Benutzer-ID, ein Benutzername und optional ein Passwort und eine Zugriffszeichenfolge. Die Zugriffszeichenfolge stellt die Berechtigungsstufe für Schlüssel und Befehle bereit. Die Benutzer-ID ist eindeutig für den Benutzer und der Benutzername wird an die Engine übergeben.

Vergewissern Sie sich, dass die von Ihnen bereitgestellten Benutzerberechtigungen für den beabsichtigten Zweck der Benutzergruppe sinnvoll sind. Wenn Sie beispielsweise eine Benutzergruppe mit dem Namen `Administrators` erstellen, sollte jeder Benutzer, den Sie dieser Gruppe hinzufügen, seine Zugriffszeichenfolge auf vollen Zugriff auf Schlüssel und Befehle gesetzt haben. Für Benutzer in einer `e-commerce`-Benutzergruppe können Sie die Zugriffszeichenfolge auf Nur-Lese-Zugriff setzen.

ElastiCache konfiguriert automatisch einen Standardbenutzer mit Benutzer-ID und Benutzername "default" und fügt ihn allen Benutzergruppen hinzu. Sie können diesen Benutzer nicht löschen oder ändern. Dieser Benutzer ist aus Kompatibilitätsgründen mit dem Standardverhalten früherer OSS Redis-Versionen vorgesehen und verfügt über eine Zugriffszeichenfolge, die es ihm ermöglicht, alle Befehle aufzurufen und auf alle Schlüssel zuzugreifen.

Wenn Sie einem Cache eine angemessene Zugriffskontrolle hinzufügen möchten, ersetzen Sie diesen Standardbenutzer durch einen neuen, der nicht aktiviert ist oder ein sicheres Passwort verwendet. Um den Standardbenutzer zu ändern, erstellen Sie einen neuen Benutzer, dessen Benutzername auf default gesetzt ist. Sie können ihn dann mit dem ursprünglichen Standardbenutzer austauschen.

Führen Sie die folgenden Schritte aus, wie Sie den ursprünglichen default-Benutzer mit einem anderen default-Benutzer mit einer geänderten Zugriffszeichenfolge getauscht haben.

Ändern Sie den Standardbenutzer in der Konsole wie folgt:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus dem Navigationsbereich Benutzergruppenverwaltung aus.
3. Wählen Sie für Benutzergruppen-ID die ID aus, die Sie ändern möchten. Stellen Sie sicher, dass Sie den Link und nicht das Kontrollkästchen ausgewählt haben.
4. Wählen Sie Ändern aus.
5. Wählen Sie im Fenster Ändern die Option Verwalten und für den Benutzer, der als Standardbenutzer festgelegt werden, Benutzername aus.
6. Wählen Sie Choose (Auswählen) aus.
7. Wählen Sie Ändern aus. Wenn Sie dies tun, werden alle vorhandenen Verbindungen mit einem Cache, die der ursprüngliche Standardbenutzer hatte, beendet.

Um den Standardbenutzer mit dem zu ändern AWS CLI

1. Erstellen Sie einen neuen Benutzer mit dem Benutzernamen default unter Verwendung der folgenden Befehle.

Für Linux, macOS oder Unix:

```
aws elasticache create-user \
```

```
--user-id "new-default-user" \
--user-name "default" \
--engine "REDIS" \
--passwords "a-strong-password" \
--access-string "off +get ~keys*"
```

Für Windows:

```
aws elasticache create-user ^
--user-id "new-default-user" ^
--user-name "default" ^
--engine "REDIS" ^
--passwords "a-strong-password" ^
--access-string "off +get ~keys*"
```

- Erstellen Sie eine Benutzergruppe und fügen Sie den von Ihnen zuvor erstellten Benutzer hinzu.

Für Linux, macOS oder Unix:

```
aws elasticache create-user-group \
--user-group-id "new-group-2" \
--engine "REDIS" \
--user-ids "new-default-user"
```

Für Windows:

```
aws elasticache create-user-group ^
--user-group-id "new-group-2" ^
--engine "REDIS" ^
--user-ids "new-default-user"
```

- Tauschen Sie den neuen default-Benutzer mit dem ursprünglichen default-Benutzer.

Für Linux, macOS oder Unix:

```
aws elasticache modify-user-group \
--user-group-id test-group \
--user-ids-to-add "new-default-user" \
--user-ids-to-remove "default"
```

Für Windows:

```
aws elasticache modify-user-group ^
 --user-group-id test-group ^
 --user-ids-to-add "new-default-user" ^
 --user-ids-to-remove "default"
```

Wenn dieser Änderungsvorgang aufgerufen wird, werden alle vorhandenen Verbindungen mit einem Cache, die der ursprüngliche Standardbenutzer hatte, beendet.

Beim Erstellen eines Benutzers können Sie bis zu zwei Passwörter einrichten. Wenn Sie ein Passwort ändern, werden alle vorhandenen Verbindungen mit Caches beibehalten.

Beachten Sie bei der Verwendung von RBAC for ElastiCache (RedisOSS) insbesondere die folgenden Einschränkungen für Benutzerkennwörter:

- Passwörter müssen 16-128 druckbare Zeichen enthalten.
- Folgende nicht-alphanumerische Zeichen sind nicht zulässig: , " " / @.

## Benutzer mit der Konsole verwalten und CLI

Gehen Sie wie unten beschrieben vor, um die Benutzer in der Konsole zu verwalten.

### Verwalten von Benutzern in der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Amazon-Dashboard Benutzerverwaltung aus. Verfügbar sind die nachfolgend aufgeführten Optionen:
  - Benutzer erstellen – Beim Erstellen eines Benutzers geben Sie eine Benutzer-ID, einen Benutzernamen, einen Authentifizierungsmodus und eine Zugriffszeichenfolge ein. Die Zugriffszeichenfolge legt die Berechtigungsstufe fest, welche Schlüssel und Befehle für den Benutzer erlaubt sind.

Beim Erstellen eines Benutzers können Sie bis zu zwei Passwörter einrichten. Wenn Sie ein Passwort ändern, werden alle vorhandenen Verbindungen mit Caches beibehalten.

- Benutzer ändern – Ermöglicht Ihnen, die Authentifizierungseinstellungen eines Benutzers zu aktualisieren oder dessen Zugriffszeichenfolge zu ändern.

- Benutzer löschen –Das Konto wird aus allen Benutzergruppen entfernt, zu denen es gehört.

Gehen Sie wie unten beschrieben vor, um die Benutzer im AWS CLI zu verwalten.

Um einen Benutzer zu ändern, verwenden Sie CLI

- Verwenden des `modify-user`-Befehls, um das Passwort oder die Passwörter eines Benutzers zu aktualisieren oder die Zugriffsberechtigungen eines Benutzers zu ändern.

Wenn ein Benutzer geändert wird, werden die Benutzergruppen, die dem Benutzer zugeordnet sind, gemeinsam mit allen Caches aktualisiert, die der Benutzergruppe zugeordnet sind. Alle vorhandenen Verbindungen werden gewartet. Im Folgenden sind einige Beispiele aufgeführt.

Für Linux, macOS oder Unix:

```
aws elasticache modify-user \
 --user-id user-id-1 \
 --access-string "~objects:* ~items:* ~public:*" \
 --no-password-required
```

Für Windows:

```
aws elasticache modify-user ^
 --user-id user-id-1 ^
 --access-string "~objects:* ~items:* ~public:*" ^
 --no-password-required
```

#### Note

Wir raten davon ab, die `nopass`-Option zu nutzen. Wenn Sie dies tun, empfehlen wir, die Berechtigungen des Benutzers auf Nur-Lese-Zugriff und auf einen begrenzten Satz von Schlüsseln festzulegen.

## Um einen Benutzer mit dem zu löschen CLI

- Verwenden Sie den Befehl `delete-user`, um einen Benutzer zu löschen. Das Konto wird gelöscht und aus allen zugehörigen Benutzergruppen entfernt. Im Folgenden wird ein Beispiel gezeigt.

Für Linux, macOS oder Unix:

```
aws elasticache delete-user \
 --user-id user-id-2
```

Für Windows:

```
aws elasticache delete-user ^
 --user-id user-id-2
```

Um eine Liste der Benutzer anzuzeigen, rufen Sie die [describe-user](#)-Operation auf.

```
aws elasticache describe-users
```

## Benutzergruppen mit der Konsole verwalten und CLI

Sie können Benutzergruppen erstellen, um den Zugriff von Benutzern auf einen oder mehrere Caches zu organisieren und zu steuern, wie im Folgenden dargestellt.

Gehen Sie wie unten beschrieben vor, um die Benutzergruppen in der Konsole zu verwalten.

So verwalten Sie Benutzergruppen mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im ElastiCache Amazon-Dashboard die Option Benutzergruppenverwaltung aus.

Die folgenden Operationen sind verfügbar, um neue Benutzergruppen zu erstellen:

- Erstellen – Wenn Sie eine Benutzergruppe erstellen, fügen Sie Benutzer hinzu und ordnen die Benutzergruppen dann Caches zu. So können Sie beispielsweise eine Admin-Benutzergruppe für Benutzer erstellen, die über Administratorrollen in einem Cache verfügen.

 **Important**

Wenn Sie eine Benutzergruppe erstellen, müssen Sie den Standardbenutzer einbeziehen.

- Benutzer hinzufügen – Fügen Sie der Benutzergruppe Benutzer hinzu.
- Entfernen von Benutzern – Entfernt Benutzer aus der Benutzergruppe. Wenn Benutzer aus einer Benutzergruppe entfernt werden, werden alle vorhandenen Verbindungen, die diese mit einem Cache haben, beendet.
- Löschen – Verwenden Sie diese Option, um eine Benutzergruppe zu löschen. Beachten Sie, dass nur die Benutzergruppe selbst und nicht die Benutzer, die zu dieser Gruppe gehören, gelöscht wird.

Für bestehende Benutzergruppen können Sie die folgenden Aktionen ausführen:

- Benutzer hinzufügen – Fügen Sie der Benutzergruppe bereits vorhandene Benutzer hinzu.
- Benutzer löschen – Entfernt vorhandene Benutzer aus der Benutzergruppe.

 **Note**

Benutzer werden von der Benutzergruppe entfernt, aber nicht aus dem System gelöscht.

Verwenden Sie die folgenden Verfahren, um Benutzergruppen mithilfe von zu verwaltenCLI.

Um eine neue Benutzergruppe zu erstellen und einen Benutzer hinzuzufügen, verwenden Sie CLI

- Geben Sie wie nachfolgend gezeigt den `create-user-group`-Befehl ein.

Für Linux, macOS oder Unix:

```
aws elasticache create-user-group \
 --user-group-id "new-group-1" \
 --engine "REDIS" \
 --user-ids user-id-1, user-id-2
```

## Für Windows:

```
aws elasticache create-user-group ^
 --user-group-id "new-group-1" ^
 --engine "REDIS" ^
 --user-ids user-id-1, user-id-2
```

Um eine Benutzergruppe zu ändern, indem Sie neue Benutzer hinzufügen oder aktuelle Mitglieder entfernen, verwenden Sie CLI

- Geben Sie wie nachfolgend gezeigt den `modify-user-group`-Befehl ein.

## Für Linux, macOS oder 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
```

## Für 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

Alle offenen Verbindungen, die zu einem Benutzer gehören, der aus einer Benutzergruppe entfernt wurde, werden mit diesem Befehl beendet.

Um eine Benutzergruppe zu löschen, verwenden Sie CLI

- Geben Sie wie nachfolgend gezeigt den `delete-user-group`-Befehl ein. Die Benutzergruppe selbst, nicht die zur Gruppe gehörenden Benutzer, wird gelöscht.

## Für Linux, macOS oder Unix:

```
aws elasticache delete-user-group /
 --user-group-id
```

Für Windows:

```
aws elasticache delete-user-group ^
 --user-group-id
```

Um eine Liste von Benutzergruppen zu sehen, können Sie den [describe-user-groups](#) Vorgang aufrufen.

```
aws elasticache describe-user-groups \
 --user-group-id test-group
```

## Zuweisen von Benutzergruppen zu Serverless-Caches

Nachdem Sie eine Benutzergruppe erstellt und Benutzer hinzugefügt haben, besteht der letzte Schritt bei der Implementierung RBAC darin, die Benutzergruppe einem serverlosen Cache zuzuweisen.

## Zuweisen von Benutzergruppen zu Serverless-Caches mithilfe der Konsole

Gehen Sie wie folgt vor, um einem serverlosen Cache mithilfe von eine Benutzergruppe hinzuzufügen: AWS Management Console

- Informationen zum deaktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- Informationen zum aktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#)

## Zuweisen von Benutzergruppen zu serverlosen Caches mithilfe von AWS CLI

Der folgende AWS CLI Vorgang erstellt einen serverlosen Cache unter Verwendung des `user-group-id` Parameters mit dem Wert `my-user-group-id`. Ersetzen Sie die Subnetzgruppe `sng-test` durch eine vorhandene Subnetzgruppe.

## Hauptparameter

- **--engine**— Muss `valkey` oder `redis` sein.

- **--user-group-id** – Dieser Wert gibt die ID der Benutzergruppe an, die aus Benutzern mit bestimmten Zugriffsberechtigungen für den Cache besteht.

Für Linux, macOS oder Unix:

```
aws elasticache create-serverless-cache \
 --serverless-cache-name "new-serverless-cache" \
 --description "new-serverless-cache" \
 --engine "redis" \
 --user-group-id "new-group-1"
```

Für Windows:

```
aws elasticache create-serverless-cache ^
 --serverless-cache-name "new-serverless-cache" ^
 --description "new-serverless-cache" ^
 --engine "redis" ^
 --user-group-id "new-group-1"
```

Die folgende AWS CLI Operation ändert einen serverlosen Cache mit dem `user-group-id` Parameter mit dem Wert `my-user-group-id`

Für Linux, macOS oder Unix:

```
aws elasticache modify-serverless-cache \
 --serverless-cache-name serverless-cache-1 \
 --user-group-id "new-group-2"
```

Für Windows:

```
aws elasticache modify-serverless-cache ^
 --serverless-cache-name serverless-cache-1 ^
 --user-group-id "new-group-2"
```

Beachten Sie, dass jegliche an einem Cache vorgenommenen Änderungen asynchron aktualisiert werden. Sie können den Fortschritt durch Ansicht der Ereignismeldungen überwachen. Weitere Informationen finden Sie unter [ElastiCache Ereignisse anzeigen](#).

## Zuweisung von Benutzergruppen zu Replikationsgruppen

Nachdem Sie eine Benutzergruppe erstellt und Benutzer hinzugefügt haben, besteht der letzte Schritt bei der Implementierung RBAC darin, die Benutzergruppe einer Replikationsgruppe zuzuweisen.

### Zuweisen von Benutzergruppen zu Replikationsgruppen mit der Konsole

Gehen Sie wie folgt vor, um einer Replikation mithilfe von eine Benutzergruppe hinzuzufügen: AWS Management Console

- Informationen zum deaktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- Informationen zum aktivierten Cluster-Modus finden Sie unter [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#)

### Zuweisen von Benutzergruppen zu Replikationsgruppen mithilfe von AWS CLI

Der folgende AWS CLI Vorgang erstellt eine Replikationsgruppe mit aktivierter Verschlüsselung bei der Übertragung (TLS) und dem `user-group-ids` Parameter mit dem Wert `my-user-group-id`. Ersetzen Sie die Subnetzgruppe `sng-test` durch eine vorhandene Subnetzgruppe.

#### Hauptparameter

- **--engine**— Muss `valkey` oder `seinredis`.
- **--engine-version** – Muss 6.0 oder höher sein.
- **--transit-encryption-enabled** – Erforderlich für die Authentifizierung und zum Zuordnen einer Benutzergruppe.
- **--user-group-ids** – Dieser Wert gibt die ID der Benutzergruppe an, die aus Benutzern mit bestimmten Zugriffsberechtigungen für den Cache besteht.
- **--cache-subnet-group** – Erforderlich für die Zuordnung einer Benutzergruppe.

Für Linux, macOS oder 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
```

```
--transit-encryption-enabled \
--user-group-ids "new-group-1" \
--cache-subnet-group "cache-subnet-group"
```

Für 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"
```

Der folgende AWS CLI Vorgang ändert eine Replikationsgruppe mit aktivierter Verschlüsselung bei der Übertragung (TLS) und dem `user-group-ids` Parameter mit dem Wert *my-user-group-id*.

Für Linux, macOS oder 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"
```

Für 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"
```

Beachten Sie die `PendingChanges` in der Antwort. Jegliche an einem Cache vorgenommenen Änderungen werden asynchron aktualisiert. Sie können den Fortschritt durch Ansicht der Ereignismeldungen überwachen. Weitere Informationen finden Sie unter [ElastiCache Ereignisse anzeigen](#).

### Migration von zu AUTH RBAC

Gehen Sie AUTH wie unter beschrieben vor, wenn Sie auf Using umsteigen möchten RBAC, gehen Sie wie folgt vor. [Authentifizierung mit den Befehlen Valkey und Redis OSS AUTH](#)

Gehen Sie wie folgt vor, um von der RBAC Konsole AUTH zur zu migrieren.

Um von Valkey oder Redis OSS AUTH zur RBAC Verwendung der Konsole zu migrieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der sich der Cache befindet, den Sie ändern möchten.
3. Wählen Sie im Navigationsbereich die Engine, die in dem Cache ausgeführt wird, den Sie ändern möchten.

Es wird eine Liste der Caches der ausgewählten Engine angezeigt.

4. Wählen Sie in der Liste der Caches den Namen des Caches aus, den Sie ändern möchten.
5. Wählen Sie für Actions (Aktionen) die Option Modify (Ändern) aus.

Das Fenster Ändern wird angezeigt.

6. Wählen Sie für Zugriffskontroll die Option Zugriffskontrollliste für Benutzergruppen aus.
7. Wählen Sie für Zugriffskontrollliste für Benutzergruppen eine Benutzergruppe aus.
8. Wählen Sie Änderungen in der Vorschau anzeigen und dann auf dem nächsten Bildschirm Ändern aus.

Gehen Sie wie folgt vor, um von Valkey oder Redis OSS AUTH zur Verwendung von zu migrieren.  
RBAC CLI

Um von zu mit dem AUTH zu RBAC migrieren CLI

- Geben Sie wie nachfolgend gezeigt den `modify-replication-group`-Befehl ein.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group --replication-group-id test \
--auth-token-update-strategy DELETE \
--user-group-ids-to-add user-group-1
```

Für Windows:

```
aws elasticache modify-replication-group --replication-group-id test ^
--auth-token-update-strategy DELETE ^
--user-group-ids-to-add user-group-1
```

## Migration von nach RBAC AUTH

Wenn Sie Redis verwenden RBAC und zu Redis OSS AUTH migrieren möchten, finden Sie weitere Informationen unter [Migration von zu RBAC AUTH](#)

### Note

Wenn Sie die Zugriffskontrolle für einen ElastiCache Cache deaktivieren müssen, müssen Sie dies über den AWS CLI tun. Weitere Informationen finden Sie unter [the section called "Deaktivieren der Zugriffskontrolle für einen ElastiCache Valkey- oder Redis-Cache OSS"](#).

## Automatisches Rotieren von Passwörtern für Benutzer

Mit AWS Secrets Manager können Sie hartcodierte Anmeldeinformationen in Ihrem Code (einschließlich Kennwörtern) durch einen API Aufruf von Secrets Manager ersetzen, um das Geheimnis programmgesteuert abzurufen. Dadurch wird sichergestellt, dass das Secret nicht kompromittiert werden kann, wenn jemand Ihren Code durchsucht, da es sich gar nicht dort befindet. Außerdem können Sie Secrets Manager so konfigurieren, dass er das Secret automatisch nach einem von Ihnen festgelegten Zeitplan rotiert. So können Sie Secrets mit langer Einsatzdauer durch Secrets mit kurzer Einsatzdauer ersetzen und damit das Risiko einer Kompromittierung erheblich verringern.

Mit Secrets Manager können Sie Ihre ElastiCache (Redis-OSS) Passwörter (d. h. Secrets) mithilfe einer von Secrets Manager AWS Lambda bereitgestellten Funktion automatisch rotieren.

Weitere Informationen zu finden Sie AWS Secrets Manager unter [Was ist AWS Secrets Manager?](#)

## Wie ElastiCache verwendet Secrets

Valkey 7.2 hat einen äquivalenten Funktionsumfang wie Redis 7.0OSS. In Redis OSS 6, ElastiCache eingeführt, [Rollenbasierte Zugriffskontrolle \(\) RBAC](#) um den Valkey- oder Redis-Cluster zu sichern. OSS Mit dieser Funktion können bestimmte Verbindungen bezüglich der ausführbaren Befehle und der Tasten, auf die zugegriffen werden kann, eingeschränkt werden. Während der Kunde einen

Benutzer mit Passwörtern erstellt, müssen die Passwortwerte manuell im Klartext eingegeben werden und sind für den Bediener sichtbar. RBAC

Mit Secrets Manager wird das Passwort von Anwendungen aus Secrets Manager abgerufen, anstatt es manuell einzugeben und in der Anwendungskonfiguration zu speichern. Weitere Informationen hierzu finden Sie unter [Wie werden ElastiCache Benutzer mit dem Geheimnis verknüpft](#).

Für die Verwendung von Secrets fallen Kosten an. Preisinformationen finden Sie unter [AWS Secrets Manager – Preise](#).

Wie werden ElastiCache Benutzer mit dem Geheimnis verknüpft

Secrets Manager speichert eine Referenz für den zugehörigen Benutzer im `SecretString`-Feld des Secrets. Von der ElastiCache Seite wird es keinen Hinweis auf das Geheimnis geben.

```
{
 "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
}
```

## Lambda-Rotationsfunktion

Um die automatische Passwortrotation von Secrets Manager zu aktivieren, erstellen Sie eine Lambda-Funktion, die mit dem [Modify-Benutzer interagiert, um die Passwörter des Benutzers](#) API zu aktualisieren.

Informationen zur Funktionsweise finden Sie unter [Funktionsweise der Drehung](#).

### Note

Für einige AWS Dienste AWS empfiehlt es sich, sowohl den Bedingungschlüssel als auch den globalen Bedingungschlüssel zu verwenden, um das Szenario eines verwirrten Stellvertreters zu vermeiden. `aws:SourceArn` `aws:SourceAccount` Wenn Sie die `aws:SourceArn` Bedingung jedoch in Ihre Rotationsfunktionsrichtlinie aufnehmen, kann die Rotationsfunktion nur verwendet werden, um das darin angegebene Geheimnis zu rotierenARN. Es wird empfohlen, nur den Kontextschlüssel `aws:SourceAccount` anzugeben, damit Sie die Drehungsfunktion für mehrere Geheimnisse verwenden können.

Informationen zu Problemen, auf die Sie möglicherweise stoßen, finden Sie unter [Fehlerbehebung bei der AWS Secrets Manager Manager-Rotation](#).

So erstellen Sie einen ElastiCache Benutzer und verknüpfen ihn mit Secrets Manager

Die folgenden Schritte veranschaulichen, wie Sie einen Benutzer erstellen und ihn mit Secrets Manager verknüpfen:

#### 1. Einen inaktiven Benutzer erstellen

Für Linux, macOS oder 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
```

Für 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
```

Es wird eine Antwort ähnlich dem folgenden Beispiel angezeigt:

```
{
 "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. Erstellen eines -Secrets

Für Linux, macOS oder Unix:

```
aws secretsmanager create-secret \
--name production/ec/user1 \
--secret-string \
'{
 "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
 "username": "user1"
}'
```

Für Windows:

```
aws secretsmanager create-secret ^
--name production/ec/user1 ^
--secret-string ^
'{
 "user_arn": "arn:aws:elasticache:us-east-1:123456xxxx:user:user1",
 "username": "user1"
}'
```

Es wird eine Antwort ähnlich dem folgenden Beispiel angezeigt:

```
{
 "ARN": "arn:aws:secretsmanager:us-east-1:123456xxxx:secret:production/ec/user1-
eaFois",
 "Name": "production/ec/user1",
 "VersionId": "aae5b963-1e6b-4250-91c6-ebd6c47d0d95"
}
```

## 3. Eine Lambda-Funktion zur Rotation Ihres Passworts konfigurieren

- a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Lambda-Konsole unter <https://console.aws.amazon.com/lambda/>
- b. Wählen Sie im Navigationsbereich Functions (Funktionen) und dann die erstellte Funktion aus. Wählen Sie den Funktionsnamen aus, nicht das Kontrollkästchen links davon.
- c. Wählen Sie die Registerkarte Konfiguration aus.

- d. Wählen Sie unter General configuration (Allgemeine Konfiguration) die Option Edit (Bearbeiten) aus und legen Sie die Timeout (Zeitüberschreitung) anschließend auf mindestens 12 Minuten fest.
- e. Wählen Sie Save (Speichern) aus.
- f. Wählen Sie Environment variables (Umgebungsvariablen) aus und legen Sie dann Folgendes fest:
  - i. SECRETS\_MANAGER\_ENDPOINT — <https://secretsmanager.REGION.amazonaws.com>
  - ii. SECRET\_ARN — Der Amazon-Ressourcenname (ARN) des Geheimnisses, das Sie in Schritt 2 erstellt haben.
  - iii. USER\_NAME — Nutzernamen des ElastiCache Benutzers,
  - iv. Wählen Sie Save (Speichern) aus.
- g. Wählen Sie Permissions (Berechtigungen) aus.
- h. Wählen Sie unter Ausführungsrolle den Namen der Lambda-Funktionsrolle aus, die auf der IAM Konsole angezeigt werden soll.
- i. Die Lambda-Funktion benötigt die folgende Berechtigung, um die Benutzer zu ändern und das Passwort festzulegen:

### ElastiCache

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:DescribeUsers",
 "elasticache:ModifyUser"
],
 "Resource": "arn:aws:elasticache:us-east-1:xxxxxxxxxx918:user:user1"
 }
]
}
```

### Secrets Manager

```
{
```

```
"Version": "2012-10-17",
"Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "secretsmanager:GetSecretValue",
 "secretsmanager:DescribeSecret",
 "secretsmanager:PutSecretValue",
 "secretsmanager:UpdateSecretVersionStage"
],
 "Resource": "arn:aws:secretsmanager:us-
east-1:xxxxxxxxxxxx:secret:XXXX"
 },
 {
 "Effect": "Allow",
 "Action": "secretsmanager:GetRandomPassword",
 "Resource": "*"
 }
]
}
```

#### 4. Die Secrets-Manager-Rotation einrichten

- a. Verwenden von AWS Management Console, siehe [Automatische Rotation für AWS Secrets Manager Manager-Geheimnisse mithilfe der Konsole einrichten](#)

Weitere Informationen zum Einrichten eines Rotationszeitplans finden Sie unter [Zeitplanausdrücke in Secrets-Manager-Rotation](#).

- b. Verwenden von AWS CLI, siehe [Automatische Rotation einrichten für die AWS Secrets Manager Verwendung von AWS Command Line Interface](#)

### Authentifizierung mit IAM

#### Themen

- [Übersicht](#)
- [Einschränkungen](#)
- [Aufstellen](#)
- [Herstellen von Verbindungen](#)

## Übersicht

Mit der IAM Authentifizierung können Sie eine Verbindung zu ElastiCache Valkey oder Redis OSS mithilfe von AWS IAM Identitäten authentifizieren, wenn Ihr Cache für die Verwendung von Valkey oder Redis Version 7 oder höher konfiguriert ist. OSS Auf diese Weise können Sie Ihr Sicherheitsmodell stärken und viele administrative Sicherheitsaufgaben vereinfachen. Sie können die IAM Authentifizierung auch verwenden, um eine differenzierte Zugriffskontrolle für jeden einzelnen ElastiCache Cache und jeden einzelnen Benutzer zu konfigurieren, wobei die Prinzipien der geringsten Zugriffsrechte eingehalten werden. ElastiCache IAMDie Authentifizierung ElastiCache mit Valkey oder Redis OSS funktioniert, indem im Befehl Valkey oder Redis oder ein kurzlebiges IAM Authentifizierungstoken anstelle eines langlebigen ElastiCache Benutzerkennworts bereitgestellt wird. OSS AUTH HELLO Weitere Informationen zum IAM Authentifizierungstoken finden Sie im AWS Allgemeinen Referenzhandbuch zum [Signaturprozess von Signature Version 4](#) und im folgenden Codebeispiel.

Sie können IAM Identitäten und die zugehörigen Richtlinien verwenden, um den Zugriff auf Valkey oder OSS Redis weiter einzuschränken. Sie können Benutzern von ihren Federated Identity-Anbietern auch direkten Zugriff auf Valkey- oder Redis-Caches gewähren. OSS

Um AWS IAM mit zu verwenden ElastiCache, müssen Sie zunächst einen ElastiCache Benutzer erstellen, für den der Authentifizierungsmodus auf eingestellt ist. IAM Anschließend können Sie eine IAM Identität erstellen oder wiederverwenden. Für die IAM Identität ist eine zugehörige Richtlinie erforderlich, um dem ElastiCache Cache und dem ElastiCache Benutzer die `elasticache:Connect` Aktion zu gewähren. Nach der Konfiguration können Sie ein IAM Authentifizierungstoken mit den AWS Anmeldeinformationen des IAM Benutzers oder der Rolle erstellen. Schließlich müssen Sie das kurzlebige IAM Authentifizierungstoken als Passwort in Ihrem Valkey- oder OSS Redis-Client angeben, wenn Sie eine Verbindung zu Ihrem Cache herstellen. Ein Valkey- oder OSS Redis-Client mit Unterstützung für den Anbieter von Anmeldeinformationen kann die temporären Anmeldeinformationen für jede neue Verbindung automatisch generieren. ElastiCache führt die IAM Authentifizierung für Verbindungsanfragen von IAM -aktivierten ElastiCache Benutzern durch und validiert die Verbindungsanfragen mit. IAM

## Einschränkungen

Bei der Verwendung der IAM Authentifizierung gelten die folgenden Einschränkungen:

- IAMDie Authentifizierung ist bei Verwendung ElastiCache mit Valkey 7.2 oder höher und Redis OSS Version 7.0 oder höher verfügbar.

- Für IAM -aktivierte ElastiCache Benutzer müssen die Eigenschaften Benutzername und Benutzer-ID identisch sein.
- Das IAM Authentifizierungstoken ist 15 Minuten gültig. Für langlebige Verbindungen empfehlen wir die Verwendung eines Valkey- oder OSS Redis-Clients, der eine Credentials-Provider-Schnittstelle unterstützt.
- Eine IAM authentifizierte Verbindung zu ElastiCache Valkey oder Redis OSS wird nach 12 Stunden automatisch unterbrochen. Die Verbindung kann um 12 Stunden verlängert werden, indem ein AUTH HELLO OR-Befehl mit einem neuen Authentifizierungstoken gesendet wird. IAM
- IAM Die Authentifizierung wird in MULTI EXEC Befehlen nicht unterstützt.
- Derzeit unterstützt die IAM Authentifizierung die folgenden globalen Bedingungskontextschlüssel:
  - Bei Verwendung der IAM Authentifizierung mit serverlosen Caches werden `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime`, und `aws:ResourceTag/%s` (von zugehörigen serverlosen Caches und Benutzern) unterstützt.
  - Bei Verwendung der IAM Authentifizierung mit Replikationsgruppen werden `aws:SourceIp` und `aws:ResourceTag/%s` (von zugehörigen Replikationsgruppen und Benutzern) unterstützt.

Weitere Informationen zu globalen Bedingungskontextschlüsseln finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM Benutzerhandbuch.

## Aufstellen

So richten Sie die IAM Authentifizierung ein:

### 1. Erstellen eines Caches

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

2. Erstellen Sie für Ihre Rolle ein Dokument mit IAM Vertrauensrichtlinien, das es Ihrem Konto ermöglicht, die neue Rolle anzunehmen, wie unten dargestellt. Speichern Sie die Richtlinie in einer Datei namens `trust-policy.json`.

```
{
 "Version": "2012-10-17",
```

```
 "Statement": {
 "Effect": "Allow",
 "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
 "Action": "sts:AssumeRole"
 }
 }
}
```

- Erstellen Sie ein IAM Richtliniendokument, wie unten gezeigt. Speichern Sie die Richtlinie in einer Datei namens `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"
]
 }
]
}
```

- Erstellen Sie eine IAM Rolle.

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

- Erstellen Sie die IAM Richtlinie.

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

- Hängen Sie die IAM Richtlinie an die Rolle an.

```
aws iam attach-role-policy \
--role-name "elasticache-iam-auth-app" \
```

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

## 7. Erstellen Sie einen neuen IAM -fähigen Benutzer.

```
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. Erstellen Sie eine Benutzergruppe und fügen Sie einen Benutzer an.

```
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
```

## Herstellen von Verbindungen

### Verbinden mit Token als Passwort

Sie müssen zuerst das kurzlebige IAM Authentifizierungstoken mithilfe einer vorsignierten [AWS SigV4-Anfrage](#) generieren. Danach geben Sie das IAM Authentifizierungstoken als Passwort an, wenn Sie eine Verbindung zu einem Valkey- oder OSS Redis-Cache herstellen, wie im folgenden Beispiel gezeigt.

```
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();
```

Im Folgenden finden Sie die Definition für `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());
 }
}
```

## Verbinden mit Anbieter von Anmeldeinformationen

Der folgende Code zeigt, wie Sie sich ElastiCache mithilfe des IAM Anbieters für Anmeldeinformationen authentifizieren.

```
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();
```

Im Folgenden finden Sie ein Beispiel für einen Lettuce OSS Redis-Client, der den `IAMAuthTokenRequest` in einen Anbieter für Anmeldeinformationen einbindet, um bei Bedarf automatisch temporäre Anmeldeinformationen zu generieren.

```
public class RedisIAMAuthCredentialsProvider 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 RedisIAMAuthCredentialsProvider(String userId,
 IAMAuthTokenRequest iamAuthTokenRequest,
 AWSCredentialsProvider awsCredentialsProvider) {
 this.userName = userId;
 this.awsCredentialsProvider = awsCredentialsProvider;
 this.iamAuthTokenRequest = iamAuthTokenRequest;
 this.iamAuthTokenSupplier =
 Suppliers.memoizeWithExpiration(this::getIamAuthToken, TOKEN_EXPIRY_SECONDS,
 TimeUnit.SECONDS);
 }

 @Override
 public Mono<RedisCredentials> resolveCredentials() {
 return Mono.just(RedisCredentials.just(userId, iamAuthTokenSupplier.get()));
 }

 private String getIamAuthToken() {
 return
 iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());
 }
}
```

## Authentifizierung mit den Befehlen Valkey und Redis OSS AUTH

### Note

Der AUTH wurde ersetzt durch [the section called “Rollenbasierte Zugriffskontrolle \(\) RBAC”](#)  
Alle serverlosen Caches müssen für die Authentifizierung verwendet werden. RBAC

Valkey- und OSS Redis-Authentifizierungstoken oder Passwörter ermöglichen es Valkey und RedisOSS, ein Passwort anzufordern, bevor Clients Befehle ausführen können, wodurch die Datensicherheit verbessert wird. Das AUTH ist nur für selbst entworfene Cluster verfügbar.

## Themen

- [Überblick über das AUTH ElastiCache mit Valkey und Redis OSS](#)
- [Anwenden der Authentifizierung auf einen ElastiCache Cluster mit Valkey oder Redis OSS](#)
- [Ändern des AUTH Tokens auf einem vorhandenen Cluster](#)
- [Migration von zu RBAC AUTH](#)

## Überblick über das AUTH ElastiCache mit Valkey und Redis OSS

Wenn Sie den AUTH ElastiCache mit Ihrem Valkey- oder OSS Redis-Cluster verwenden, gibt es einige Verbesserungen.

Beachten Sie insbesondere diese AUTH Token- oder Passwortbeschränkungen, wenn Sie Folgendes verwenden: AUTH

- Token oder Kennwörter müssen 16-128 druckbare Zeichen enthalten.
- Nicht alphanumerische Zeichen sind auf (!, &, #, \$, ^, <, >, -).
- AUTH kann nur für die Verschlüsselung während der Übertragung aktiviert werden, die ElastiCache mit Valkey- oder OSS Redis-Clustern aktiviert ist.

Um ein starkes Token einzurichten, wird empfohlen, eine strenge Passwortrichtlinie zu befolgen, z. B. die Folgende:

- Token oder Passwörter müssen mindestens drei der folgenden Zeichentypen enthalten:
  - Uppercase characters
  - Lowercase characters
  - Ziffern
  - Nicht-alphanumerische Zeichen (!, &, #, \$, ^, <, >, -)
- Token oder Passwörter dürfen kein Wörterbuchwort oder ein leicht modifiziertes Wörterbuchwort enthalten.
- Tokens oder Passwörter dürfen nicht mit einem kürzlich verwendeten Token identisch oder diesem ähnlich sein.

## Anwenden der Authentifizierung auf einen ElastiCache Cluster mit Valkey oder Redis OSS

Sie können verlangen, dass Benutzer ein Token (Passwort) auf einem tokengeschützten Valkey- oder Redis-Server eingeben. OSS Fügen Sie dazu den Parameter `--auth-token` (API:AuthToken) mit dem richtigen Token hinzu, wenn Sie Ihre Replikationsgruppe oder Ihren Replikationscluster erstellen. Fügen Sie ihn auch in alle nachfolgenden Befehle an die Replikationsgruppe oder den Cluster ein.

Der folgende AWS CLI Vorgang erstellt eine Replikationsgruppe mit aktivierter Verschlüsselung bei der Übertragung (TLS) und dem AUTH Token *This-is-a-sample-token*. Ersetzen Sie die Subnetzgruppe `sng-test` durch eine vorhandene Subnetzgruppe.

### Hauptparameter

- **--engine**— Muss `valkey` oder `redis` sein.
- **--engine-version**— Wenn die Engine Redis ist, muss es 3.2.6, 4.0.10 oder höher sein.
- **--transit-encryption-enabled**— Erforderlich für Authentifizierung und Eignung. HIPAA
- **--auth-token**— Für die HIPAA Teilnahmeberechtigung erforderlich. Dieser Wert muss das richtige Token für diesen tokengeschützten Valkey- oder Redis-Server sein. OSS
- **--cache-subnet-group**— Für die Teilnahmeberechtigung erforderlich. HIPAA

Für Linux, macOS oder 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
```

Für 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
```

## Ändern des AUTH Tokens auf einem vorhandenen Cluster

Um die Aktualisierung Ihrer Authentifizierung zu vereinfachen, können Sie das auf einem Cluster verwendete AUTH Token ändern. Sie können diese Änderung vornehmen, wenn die Engine-Version Valkey 7.2 oder höher oder Redis 5.0.6 oder höher ist. ElastiCache muss außerdem die Verschlüsselung bei der Übertragung aktiviert sein.

Das Ändern des Authentifizierungstokens unterstützt zwei Strategien: ROTATE und SET. Die ROTATE Strategie fügt dem Server ein zusätzliches AUTH Token hinzu, wobei das vorherige Token beibehalten wird. Die SET Strategie aktualisiert den Server so, dass er nur ein einzelnes AUTH Token unterstützt. Führen Sie diese Änderungsaufrufe mit dem `--apply-immediately`-Parameter durch, um die Änderungen sofort anzuwenden.

## Rotation des AUTH Tokens

Um einen Valkey- oder OSS Redis-Server mit einem neuen AUTH Token zu aktualisieren, rufen Sie den `ModifyReplicationGroup` API mit dem `--auth-token` Parameter als neuem AUTH Token und `--auth-token-update-strategy` mit dem Wert auf. ROTATE Nach Abschluss der ROTATE Änderung unterstützt der Cluster zusätzlich zu dem im Parameter angegebenen AUTH Token das vorherige Token. Wenn vor der Token-Rotation kein AUTH Token für die Replikationsgruppe konfiguriert wurde, unterstützt der Cluster das im `--auth-token` Parameter angegebene AUTH Token zusätzlich zur Unterstützung von Verbindungen ohne Authentifizierung. Informationen [Einstellung des Tokens AUTH](#) zur Aktualisierung des erforderlichen AUTH Tokens finden Sie unter Verwendung der Aktualisierungsstrategie SET.

### Note

Wenn Sie das AUTH Token nicht zuvor konfiguriert haben, unterstützt der Cluster nach Abschluss der Änderung kein anderes AUTH Token als das im Parameter `auth-token` angegebene.

Wenn diese Änderung auf einem Server durchgeführt wird, der bereits zwei AUTH Token unterstützt, wird bei diesem Vorgang auch das älteste AUTH Token entfernt. Auf diese Weise kann ein Server bis zu zwei aktuelle AUTH Token gleichzeitig unterstützen.

An diesem Punkt können Sie fortfahren, indem Sie den Client so aktualisieren, dass er das neueste AUTH Token verwendet. Nachdem die Clients aktualisiert wurden, können Sie die SET Strategie für die AUTH Token-Rotation (die im folgenden Abschnitt erklärt wird) verwenden, um ausschließlich das neue Token zu verwenden.

Mit dem folgenden AWS CLI Vorgang wird eine Replikationsgruppe so geändert, dass das AUTH Token *This-is-the-rotated-token* rotiert wird.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-rotated-token \
--auth-token-update-strategy ROTATE \
--apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^
--replication-group-id authtestgroup ^
--auth-token This-is-the-rotated-token ^
--auth-token-update-strategy ROTATE ^
--apply-immediately
```

## Einstellung des Tokens AUTH

Um einen Valkey- oder OSS Redis-Server so zu aktualisieren, dass er ein einzelnes erforderliches AUTH Token unterstützt, rufen Sie den ModifyReplicationGroup API Vorgang mit dem `--auth-token` Parameter auf, der denselben Wert wie das letzte AUTH Token hat, und dem `--auth-token-update-strategy` Parameter mit dem Wert `SET`. Die SET Strategie kann nur mit einem Cluster verwendet werden, der über 2 AUTH Token oder 1 optionales AUTH Token aus einer früheren ROTATE Strategie verfügt. Nach Abschluss der Änderung unterstützt der Server nur das im Parameter `AUTH auth-token` angegebene Token.

Mit dem folgenden AWS CLI Vorgang wird eine Replikationsgruppe geändert, auf die das AUTH Token festgelegt werden soll. *This-is-the-set-token*

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
--replication-group-id authtestgroup \
--auth-token This-is-the-set-token \
--auth-token-update-strategy SET \
--apply-immediately
```

Für Windows:

```
aws elasticache modify-replication-group ^
--replication-group-id authtestgroup ^
--auth-token This-is-the-set-token ^
--auth-token-update-strategy SET ^
--apply-immediately
```

Aktivierung der Authentifizierung auf einem vorhandenen Cluster

Rufen Sie den Vorgang auf, um die Authentifizierung auf einem vorhandenen Valkey- oder OSS Redis-Server zu aktivieren. `ModifyReplicationGroup` API Rufen Sie `ModifyReplicationGroup` mit dem `--auth-token` Parameter als neuem Token und dem `--auth-token-update-strategy` mit dem Wert auf. `ROTATE`

Nach Abschluss der `ROTATE` Änderung unterstützt der Cluster das im `--auth-token` Parameter angegebene AUTH Token und unterstützt zusätzlich die Verbindung ohne Authentifizierung. Sobald alle Client-Anwendungen aktualisiert wurden, um sich OSS mit dem AUTH Token bei Valkey oder Redis zu authentifizieren, verwenden Sie die `SET` Strategie, um das AUTH Token als erforderlich zu markieren. Die Aktivierung der Authentifizierung wird nur auf Valkey- und OSS Redis-Servern unterstützt, auf denen die Verschlüsselung bei der Übertragung () aktiviert ist. TLS

Migration von zu RBAC AUTH

Wenn Sie Benutzer mit Valkey oder Redis OSS Role-Based Access Control (RBAC) authentifizieren, wie unter beschrieben [Rollenbasierte Zugriffskontrolle \(\) RBAC](#), und Sie zu dieser migrieren möchten AUTH, verwenden Sie die folgenden Verfahren. Sie können die Migration entweder über die Konsole oder durchführen. CLI

Um von zu AUTH mithilfe der Konsole RBAC zu migrieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.

2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, in der sich der Cluster befindet, den Sie ändern möchten.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, den Sie modifizieren möchten.

Es wird eine Liste der ausgewählten Engine-Cluster angezeigt.

4. Wählen Sie in der Liste der Cluster den Namen des Clusters aus, den Sie modifizieren möchten..
5. Wählen Sie für Actions (Aktionen) die Option Modify (Ändern) aus.

Das Fenster Ändern wird angezeigt.

6. Wählen Sie für die Zugriffskontrolle AUTH Valkey-Standardbenutzerzugriff oder OSSAUTHRedis-Standardbenutzerzugriff.
7. Legen Sie unter AUTHValkey-Token oder OSSAUTHRedis-Token ein neues Token fest.
8. Wählen Sie Änderungen in der Vorschau anzeigen und dann auf dem nächsten Bildschirm Ändern aus.

Um von zu zu migrierenRBAC, verwenden Sie AUTH AWS CLI

Verwenden Sie einen der folgenden Befehle, um ein neues optionales AUTH Token für Ihre Valkey- oder OSS Redis-Replikationsgruppe zu konfigurieren. Beachten Sie, dass ein optionales Auth-Token einen nicht authentifizierten Zugriff auf die Replikationsgruppe ermöglicht, bis das Auth-Token als erforderlich gekennzeichnet ist. Verwenden Sie dabei die Aktualisierungsstrategie im folgenden Schritt. SET

Für Linux, macOS oder 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
```

Für 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
```

Nachdem Sie den obigen Befehl ausgeführt haben, können Sie Ihre Valkey- oder OSS Redis-Anwendungen aktualisieren, um sich mit dem neu konfigurierten optionalen Token bei der ElastiCache Replikationsgruppe zu authentifizieren. AUTH Verwenden Sie die Aktualisierungsstrategie SET im nachfolgenden Befehl unten, um die Auth-Token-Rotation abzuschließen. Dadurch wird das optionale AUTH Token als erforderlich markiert. Wenn die Aktualisierung des Auth-Tokens abgeschlossen ist, wird der Status der Replikationsgruppe als angezeigt ACTIVE und für alle Verbindungen zu dieser Replikationsgruppe ist eine Authentifizierung erforderlich.

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id test \
 --auth-token This-is-a-sample-token \
 --auth-token-update-strategy SET \
 --apply-immediately
```

Für 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
```

Weitere Informationen finden Sie unter [Authentifizierung mit den Befehlen Valkey und Redis OSS AUTH](#).

#### Note

Informationen dazu, wie Sie die Zugriffskontrolle auf einem ElastiCache Cluster deaktivieren müssen, finden Sie unter [the section called “Deaktivieren der Zugriffskontrolle für einen ElastiCache Valkey- oder Redis-Cache OSS”](#).

## Deaktivieren der Zugriffskontrolle für einen ElastiCache Valkey- oder Redis-Cache OSS

Folgen Sie den nachstehenden Anweisungen, um die Zugriffskontrolle für einen Valkey- oder Redis-Cache zu deaktivieren. OSS TLS Ihr Cache wird eine von zwei verschiedenen Konfigurationstypen haben: AUTH Standardbenutzerzugriff oder Zugriffskontrollliste für Benutzergruppen (). RBAC Wenn Ihr Cache mit der AUTH Konfiguration erstellt wurde, müssen Sie ihn auf die RBAC Konfiguration ändern, bevor Sie den Cache deaktivieren können, indem Sie die Benutzergruppen entfernen. Wenn Ihr Cache mit der RBAC Konfiguration erstellt wurde, können Sie ihn direkt deaktivieren.

Um einen OSS serverlosen Valkey- oder Redis-Cache zu deaktivieren, der mit konfiguriert ist RBAC

1. Entfernen Sie die Benutzergruppen, um die Zugriffskontrolle zu deaktivieren.

```
aws elasticache modify-serverless-cache --serverless-cache-name <serverless-cache>
--remove-user-group
```

2. (Optional) Stellen Sie sicher, dass dem Serverless-Cache keine Benutzergruppen zugeordnet sind.

```
aws elasticache describe-serverless-caches --serverless-cache-name <serverless-
cache>
{
 "...",
 "UserGroupId": ""
 "...",
}
```

Um einen Valkey- oder OSS Redis-Cache zu deaktivieren, der mit einem Token konfiguriert ist AUTH

1. Ändern Sie das AUTH Token in RBAC und geben Sie eine Benutzergruppe an, die hinzugefügt werden soll.

```
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. Stellen Sie sicher, dass das AUTH Token deaktiviert und eine Benutzergruppe hinzugefügt wurde.

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-id-value>
{
 "...
 "AuthTokenEnabled": false,
 "UserGroupIds": [
 "<user-group-value>"
]
 "...
}
```

3. Entfernen Sie die Benutzergruppen, um die Zugriffskontrolle zu deaktivieren.

```
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. (Optional) Stellen Sie sicher, dass dem Cluster keine Benutzergruppen zugeordnet sind. Das Feld `AuthTokenEnabled` sollte auch „false“ lauten.

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false
```

Um einen Valkey- oder OSS Redis-Cluster zu deaktivieren, der konfiguriert ist mit RBAC

1. Entfernen Sie die Benutzergruppen, um die Zugriffskontrolle zu deaktivieren.

```
aws elasticache modify-replication-group --replication-group-id <replication-group-
value> --user-group-ids-to-remove <user-group-value>
```

```
{
 "...
 "PendingModifiedValues": {
 "UserGroups": {
 "UserGroupIdsToAdd": [],
 "UserGroupIdsToRemove": [
 "<user-group-value>"
]
 }
 }
 "...
}
```

2. (Optional) Stellen Sie sicher, dass dem Cluster keine Benutzergruppen zugeordnet sind. Das Feld `AuthTokenEnabled` sollte auch „false“ lauten.

```
aws elasticache describe-replication-groups --replication-group-id <replication-
group-value>
"AuthTokenEnabled": false
```

## Richtlinie für den Datenverkehr zwischen Netzwerken

Amazon ElastiCache verwendet die folgenden Techniken, um Ihre Cache-Daten zu sichern und vor unbefugtem Zugriff zu schützen:

- [Amazon VPCs und ElastiCache Sicherheit](#) erklärt den Typ der Sicherheitsgruppe, die Sie für Ihre Installation benötigen.
- [Identity and Access Management für Amazon ElastiCache](#) Verwenden Sie zum Erteilen und Beschränken von Aktionen von Benutzern, Gruppen und Rollen.

### Themen

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [ElastiCache API und VPC Schnittstellenendpunkte \(AWS PrivateLink\)](#)
- [Subnetze und Subnetzgruppen](#)

## Amazon VPCs und ElastiCache Sicherheit

Da Datensicherheit wichtig ist, ElastiCache können Sie damit kontrollieren, wer Zugriff auf Ihre Daten hat. Wie Sie den Zugriff auf Ihre Daten kontrollieren, hängt davon ab, ob Sie Ihre Cluster in einer Amazon Virtual Private Cloud (AmazonVPC) oder Amazon EC2 -Classic gestartet haben oder nicht.

### Important

Wir haben die Verwendung von Amazon EC2 -Classic zum Starten von Clustern eingestellt. ElastiCache Alle aktuellen Generierungsknoten werden ausschließlich in Amazon Virtual Private Cloud gestartet.

Der Amazon Virtual Private Cloud (AmazonVPC) -Service definiert ein virtuelles Netzwerk, das einem herkömmlichen Rechenzentrum sehr ähnlich ist. Wenn Sie Ihr Amazon konfigurieren, können VPC Sie seinen IP-Adressbereich auswählen, Subnetze erstellen und Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen konfigurieren. Sie können dem virtuellen Netzwerk auch einen Cache-Cluster hinzufügen und den Zugriff auf den Cache-Cluster mithilfe von VPC Amazon-Sicherheitsgruppen steuern.

In diesem Abschnitt wird erklärt, wie Sie einen ElastiCache Cluster in einem Amazon manuell konfigurierenVPC. Diese Informationen richten sich an Benutzer, die ein tieferes Verständnis dafür wünschen, wie Amazon ElastiCache und Amazon VPC zusammenarbeiten.

### Themen

- [Understanding ElastiCache und Amazon VPCs](#)
- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC](#)
- [Eine virtuelle private Cloud erstellen \(VPC\)](#)
- [Verbindung zu einem Cache herstellen, der in einem Amazon läuft VPC](#)

## Understanding ElastiCache und Amazon VPCs

ElastiCache ist vollständig in die Amazon Virtual Private Cloud (AmazonVPC) integriert. Für ElastiCache Benutzer bedeutet das Folgendes:

- Wenn Ihr AWS Konto nur die VPC Plattform EC2 - unterstützt, wird Ihr Cluster ElastiCache immer in einem Amazon gestartetVPC.
- Wenn Sie noch nicht damit vertraut sind AWS, werden Ihre Cluster in einem Amazon bereitgestelltVPC. Ein Standard VPC wird automatisch für Sie erstellt.
- Wenn Sie einen Standard haben VPC und beim Starten eines Clusters kein Subnetz angeben, wird der Cluster in Ihrem Standard-Amazon VPC gestartet.

Weitere Informationen finden Sie unter [Erkennen Ihrer unterstützten Plattformen und ob Sie über eine VPC Standardplattform verfügen](#).

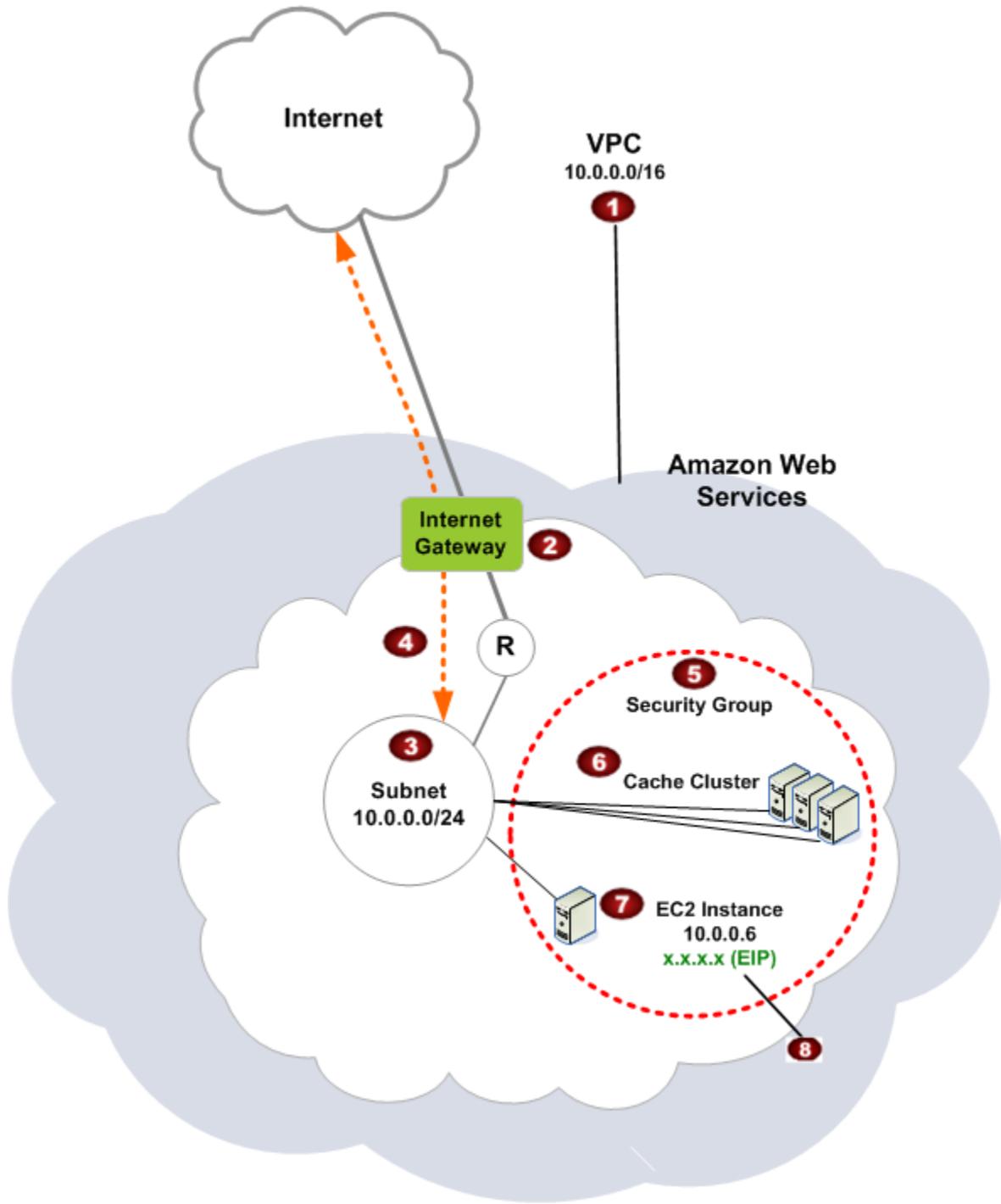
Mit Amazon Virtual Private Cloud können Sie ein virtuelles Netzwerk in der AWS Cloud erstellen, das einem herkömmlichen Rechenzentrum sehr ähnlich ist. Sie können Ihren Amazon konfigurierenVPC, einschließlich der Auswahl seines IP-Adressbereichs, der Erstellung von Subnetzen und der Konfiguration von Routentabellen, Netzwerk-Gateways und Sicherheitseinstellungen.

Die grundlegende Funktionalität von ElastiCache ist dieselbe wie in einer Virtual Private Cloud; ElastiCache verwaltet Software-Upgrades, Patches, Fehlererkennung und Wiederherstellung, unabhängig davon, ob Ihre Cluster innerhalb oder außerhalb eines Amazon VPC bereitgestellt werden.

ElastiCache Cache-Knoten, die außerhalb eines Amazon bereitgestellt VPC werden, erhalten eine IP-Adresse, auf die der DNS Endpunkt/Name aufgelöst wird. Dies bietet Konnektivität von Amazon Elastic Compute Cloud (AmazonEC2) -Instances. Wenn Sie einen ElastiCache Cluster in einem VPC privaten Amazon-Subnetz starten, wird jedem Cache-Knoten eine private IP-Adresse innerhalb dieses Subnetzes zugewiesen.

### Überblick über ElastiCache In an Amazon VPC

Das folgende Diagramm und die folgende Tabelle beschreiben die VPC Amazon-Umgebung zusammen mit ElastiCache Clustern und EC2 Amazon-Instances, die in Amazon gestartet werdenVPC.



1

Amazon VPC ist ein isolierter Teil der AWS Cloud, dem ein eigener Block von IP-Adressen zugewiesen wird.

2

Ein Internet-Gateway verbindet Ihren Amazon VPC direkt mit dem Internet und bietet Zugriff auf andere AWS Ressourcen wie Amazon Simple Storage Service (Amazon S3), die außerhalb Ihres Amazons laufenVPC.

**3** Ein VPC Amazon-Subnetz ist ein Segment des IP-Adressbereichs eines Amazon, VPC in dem Sie AWS Ressourcen entsprechend Ihren Sicherheits- und Betriebsanforderungen isolieren können.

**4** Eine Routing-Tabelle im Amazon VPC leitet den Netzwerkverkehr zwischen dem Subnetz und dem Internet. Amazon VPC hat einen impliziten Router, der in diesem Diagramm durch den Kreis mit dem R symbolisiert wird.

**5** Eine VPC Amazon-Sicherheitsgruppe kontrolliert den ein- und ausgehenden Datenverkehr für Ihre ElastiCache Cluster und EC2 Amazon-Instances.

**6** Sie können einen ElastiCache Cluster im Subnetz starten. Die Cache-Knoten haben private IP-Adressen aus dem Adressbereich des Subnetzes.

**7** Sie können EC2 Amazon-Instances auch im Subnetz starten. Jede EC2 Amazon-Instance hat eine private IP-Adresse aus dem Adressbereich des Subnetzes. Die EC2 Amazon-Instance kann eine Verbindung zu jedem Cache-Knoten im selben Subnetz herstellen.

**8** Damit eine EC2 Amazon-Instance in Ihrem Amazon VPC vom Internet aus erreichbar ist, müssen Sie der Instance eine statische, öffentliche Adresse zuweisen, die als Elastic IP-Adresse bezeichnet wird.

## Voraussetzungen

Um einen ElastiCache Cluster innerhalb eines Amazon zu erstellenVPC, VPC muss Ihr Amazon die folgenden Anforderungen erfüllen:

- Amazon VPC muss nicht dedizierte EC2 Amazon-Instances zulassen. Sie können es nicht ElastiCache in einem Amazon verwendenVPC, das für Dedicated Instance Tenancy konfiguriert ist.

- Für Ihr Amazon VPC muss eine Cache-Subnetzgruppe definiert werden. ElastiCache verwendet diese Cache-Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren VPC Endpunkten oder Cache-Knoten zugeordnet werden sollen.
- CIDR Die Blöcke für jedes Subnetz müssen groß genug sein, um Reserve-IP-Adressen für ElastiCache Wartungsarbeiten bereitzustellen.

## Weiterleitung und Sicherheit

Sie können das Routing in Ihrem Amazon konfigurieren VPC, um zu kontrollieren, wohin der Datenverkehr fließt (z. B. zum Internet-Gateway oder Virtual Private Gateway). Mit einem Internet-Gateway VPC hat Ihr Amazon direkten Zugriff auf andere AWS Ressourcen, die nicht in Ihrem Amazon laufen VPC. Wenn Sie sich dafür entscheiden, nur ein virtuelles privates Gateway mit einer Verbindung zum lokalen Netzwerk Ihrer Organisation zu verwenden, können Sie Ihren Internetverkehr über das Netzwerk weiterleiten und lokale Sicherheitsrichtlinien VPN und Firewalls verwenden, um den ausgehenden Datenverkehr zu kontrollieren. In diesem Fall fallen zusätzliche Bandbreitengebühren an, wenn Sie über das Internet auf AWS Ressourcen zugreifen.

Sie können VPC Amazon-Sicherheitsgruppen verwenden, um die ElastiCache Cluster und EC2 Amazon-Instances in Ihrem Amazon zu sichern VPC. Sicherheitsgruppen wirken wie eine Firewall auf der Instance-Ebene, nicht auf der Subnetzebene.

### Note

Wir empfehlen dringend, DNS Namen zu verwenden, um eine Verbindung zu Ihren Cache-Knoten herzustellen, da sich die zugrunde liegende IP-Adresse ändern kann.

## VPC Amazon-Dokumentation

Amazon VPC verfügt über eine eigene Dokumentation, in der beschrieben wird, wie Sie Ihr Amazon erstellen und verwenden VPC. Die folgende Tabelle enthält Links zu den VPC Amazon-Leitfäden.

Beschreibung	Dokumentation
Erste Schritte mit Amazon VPC	<a href="#">Erste Schritte mit Amazon VPC</a>
So nutzen Sie Amazon VPC über die AWS Management Console	<a href="#">VPC Amazon-Benutzerhandbuch</a>

Beschreibung	Dokumentation
Vollständige Beschreibungen aller VPC Amazon-Befehle	<a href="#">EC2Amazon-Befehlszeilenreferenz</a> (die VPC Amazon-Befehle finden Sie in der EC2 Amazon-Referenz)
Vollständige Beschreibungen der VPC API Abläufe, Datentypen und Fehler von Amazon	<a href="#">EC2APIAmazon-Referenz</a> (die VPC API Amazon-Operationen finden Sie in der EC2 Amazon-Referenz)
Informationen für den Netzwerkadministrator, der das Gateway an Ihrem Ende einer optionalen IPsec VPN Verbindung konfigurieren muss	<a href="#">Was ist AWS Site-to-Site VPN?</a>

Weitere Informationen zur Amazon Virtual Private Cloud finden Sie unter [Amazon Virtual Private Cloud](#).

## Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC

Amazon ElastiCache unterstützt die folgenden Szenarien für den Zugriff auf einen Cache in einem AmazonVPC:

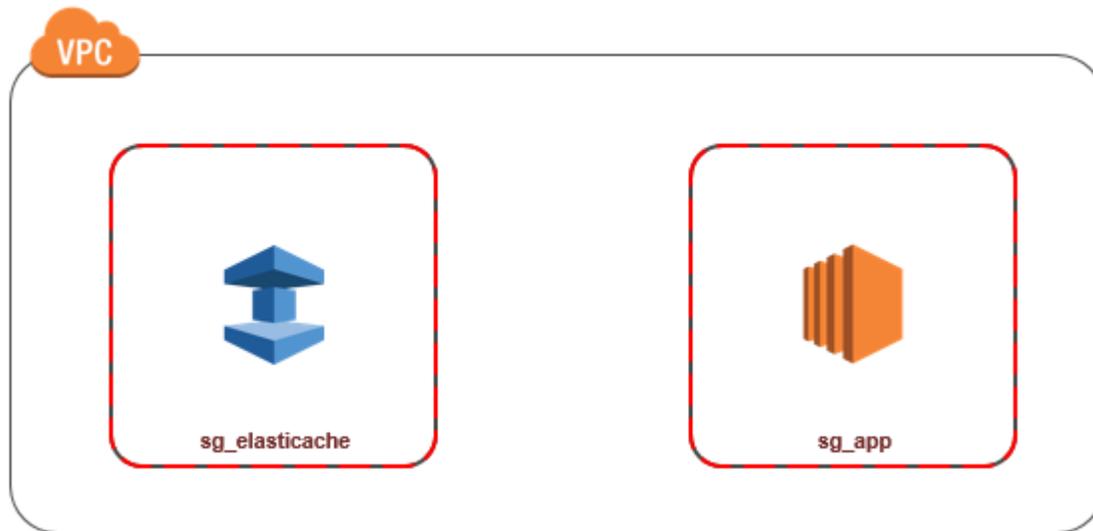
### Inhalt

- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance im selben Amazon befinden VPC](#)
- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon befinden VPCs](#)
  - [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon VPCs in derselben Region befinden](#)
    - [Verwenden von Transit Gateway](#)
  - [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in unterschiedlichen VPCs Amazon-Regionen befinden](#)
    - [Transit verwenden VPC](#)
- [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird](#)
  - [Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mithilfe von VPN Konnektivität ausgeführt wird](#)
  - [Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mit Direct Connect ausgeführt wird](#)

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance im selben Amazon befinden VPC

Der häufigste Anwendungsfall ist, wenn eine auf einer EC2 Instance bereitgestellte Anwendung eine Verbindung zu einem Cache in derselben Instanz herstellen mussVPC.

Das folgende Diagramm veranschaulicht dieses Szenario.



Der einfachste Weg, den Zugriff zwischen EC2 Instanzen und Caches gleichzeitig zu verwalten, VPC besteht darin, wie folgt vorzugehen:

1. Erstellen Sie eine VPC Sicherheitsgruppe für Ihren Cache. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf den Cache zu beschränken. Sie können beispielsweise eine benutzerdefinierte Regel für diese Sicherheitsgruppe erstellen, die den TCP Zugriff über den Port, den Sie dem Cache bei der Erstellung zugewiesen haben, und über eine IP-Adresse, die Sie für den Zugriff auf den Cache verwenden, ermöglicht.

Der Standard-Port für Memcached-Caches ist 11211.

Der Standardport für Valkey- und OSS Redis-Caches ist 6379

2. Erstellen Sie eine VPC Sicherheitsgruppe für Ihre EC2 Instanzen (Web- und Anwendungsserver). Diese Sicherheitsgruppe kann bei Bedarf den Zugriff auf die EC2 Instanz aus dem Internet über die Routingtabelle VPC von ermöglichen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, um den TCP Zugriff auf die EC2 Instance über Port 22 zu ermöglichen.
3. Erstellen Sie in der Sicherheitsgruppe für Ihren Cache benutzerdefinierte Regeln, die Verbindungen von der Sicherheitsgruppe aus zulassen, die Sie für Ihre EC2 Instances erstellt haben. Damit wird jedem Mitglied der Sicherheitsgruppe der Zugriff auf die Caches gestattet.

### Note

Wenn Sie planen, [Local Zones](#) zu nutzen, stellen Sie sicher, dass Sie sie aktiviert haben. Wenn Sie eine Subnetzgruppe in dieser lokalen Zone erstellen, VPC wird Ihre auf diese lokale Zone ausgedehnt und Sie VPC behandeln das Subnetz wie jedes Subnetz in einer anderen Availability Zone. Alle relevanten Gateways und Routing-Tabellen werden automatisch angepasst.

Um eine Regel in einer Sicherheitsgruppe zu erstellen, die Verbindungen von einer anderen VPC Sicherheitsgruppe aus zulässt

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die VPC Amazon-Konsole unter <https://console.aws.amazon.com/vpc>.
2. Wählen Sie im Navigationsbereich Security Groups (Sicherheitsgruppen) aus.
3. Wählen Sie eine Sicherheitsgruppe aus oder erstellen Sie eine, die Sie für Ihren Cache verwenden werden. Wählen Sie unter Inbound Rules (Eingangsregeln) die Option Edit Inbound Rules (Eingangsregeln bearbeiten) und dann Add Rule (Regeln hinzufügen). Diese Sicherheitsgruppe gewährt Mitgliedern einer anderen Sicherheitsgruppe Zugriff.
4. Wählen Sie unter Typ die Option Benutzerdefinierte TCP Regel aus.
  - a. Geben Sie für Port-Bereich den Port an, den Sie beim Erstellen des Caches verwendet haben.

Der Standard-Port für Memcached-Caches ist 11211.

Der Standardport für Valkey- und OSS Redis-Caches und Replikationsgruppen ist 6379

- b. Geben Sie in das Feld Source die ersten Zeichen der ID der Sicherheitsgruppe ein. Wählen Sie aus der Liste die Sicherheitsgruppe aus, die Sie für Ihre EC2 Amazon-Instances verwenden möchten.
5. Wählen Sie Save, wenn Sie fertig sind.

	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	-

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon befinden VPCs

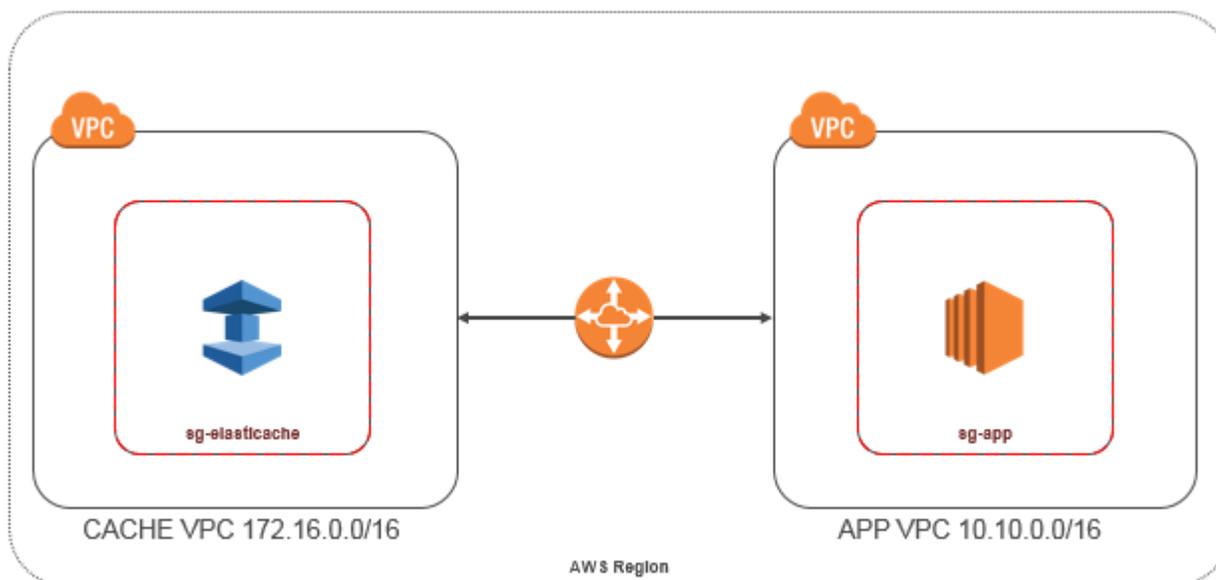
Wenn sich Ihr Cache in einer anderen als VPC der EC2 Instance befindet, die Sie für den Zugriff verwenden, gibt es mehrere Möglichkeiten, auf den Cache zuzugreifen. Wenn sich der Cache und die EC2 Instance in verschiedenen, VPCs aber in derselben Region befinden, können Sie VPC Peering verwenden. Wenn sich der Cache und die EC2 Instance in unterschiedlichen Regionen befinden, können Sie VPN Konnektivität zwischen Regionen herstellen.

## Themen

- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon VPCs in derselben Region befinden](#)
- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in unterschiedlichen VPCs Amazon-Regionen befinden](#)

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in einem anderen Amazon VPCs in derselben Region befinden

Das folgende Diagramm zeigt den Zugriff auf einen Cache durch eine EC2 Amazon-Instance in einem anderen Amazon VPC in derselben Region über eine VPC Amazon-Peering-Verbindung.



Cache, auf den eine EC2 Amazon-Instance in einem anderen Amazon VPC innerhalb derselben Region zugreift — VPC Peering-Verbindung

Eine VPC Peering-Verbindung ist eine Netzwerkverbindung zwischen zwei VPCs, die es Ihnen ermöglicht, den Verkehr zwischen ihnen mithilfe privater IP-Adressen weiterzuleiten. Instanzen in beiden VPC können miteinander kommunizieren, als ob sie sich im selben Netzwerk befinden würden. Sie können eine VPC Peering-Verbindung zwischen Ihrem eigenen Amazon VPCs oder mit einem Amazon VPC in einem anderen AWS Konto innerhalb einer einzelnen Region herstellen. Weitere Informationen zu Amazon VPC Peering finden Sie in der [VPCDokumentation](#).

### Note

Die Namensauflösung kann für Peered VPCs, abhängig von den Konfigurationen, die auf den angewendet wurden. ElastiCache VPC Um dieses Problem zu lösen, VPCs müssen beide Optionen für DNS Hostnamen und DNS Auflösung aktiviert sein. Weitere Informationen finden Sie unter [Aktivieren der DNS Auflösung für eine VPC Peering-Verbindung](#).

Um VPC über Peering auf einen Cache in einem anderen Amazon zuzugreifen

1. Stellen Sie sicher, dass sich der IP-Bereich der beiden VPCs nicht überschneidet. Andernfalls können Sie sie nicht miteinander verbinden.
2. Schauen Sie sich die beiden VPCs an. Weitere Informationen finden Sie unter [Eine Amazon VPC Peering-Verbindung erstellen und akzeptieren](#).
3. Aktualisieren Sie Ihre Routing-Tabelle. Weitere Informationen finden Sie unter [Aktualisieren Ihrer Routing-Tabellen für eine VPC Peering-Verbindung](#)

Es folgt eine Darstellung der Routing-Tabellen für das Beispiel im voranstehenden Diagramm. Es ist zu beachten, dass pcx-a894f1c1 eine Peering-Verbindung ist.

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 Routing-Tabelle

4. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches so, dass eingehende Verbindungen von der Sicherheitsgruppe der Anwendung im VPC Peering zugelassen werden. Weitere Informationen finden Sie unter [VPCReferenz-Peer-Sicherheitsgruppen](#).

Beim Zugriff auf einen Cache über eine Peering-Verbindung fallen zusätzliche Datenübertragungskosten an.

### Verwenden von Transit Gateway

Ein Transit-Gateway ermöglicht es Ihnen VPCs, VPN Verbindungen in derselben AWS Region herzustellen und den Verkehr zwischen ihnen weiterzuleiten. Ein Transit-Gateway funktioniert AWS kontenübergreifend, und Sie können AWS Resource Access Manager verwenden, um Ihr Transit-Gateway mit anderen Konten zu teilen. Nachdem Sie ein Transit-Gateway mit einem anderen AWS Konto gemeinsam genutzt haben VPCs, kann der Kontoinhaber dieses Konto mit Ihrem Transit-Gateway verknüpfen. Benutzer in einem der Konten können die Anhang jederzeit löschen.

Sie können Multicast auf einem Transit-Gateway aktivieren und dann eine Transit-Gateway-Multicast-Domäne erstellen, die es ermöglicht, Multicast-Verkehr von Ihrer Multicast-Quelle an Multicast-Gruppenmitglieder über VPC Anlagen zu senden, die Sie der Domäne zuordnen.

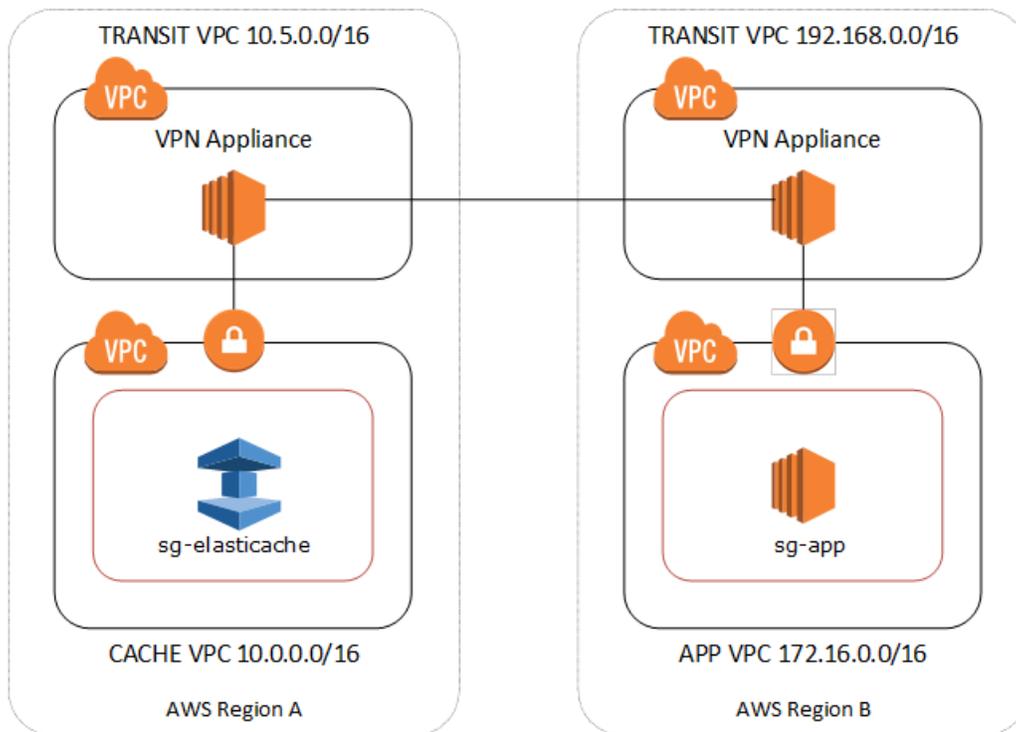
Sie können auch eine Peering-Verbindung zwischen Transit-Gateways in verschiedenen Regionen erstellen. AWS Auf diese Weise können Sie den Datenverkehr zwischen den Anhängen der Transit Gateways über verschiedene Regionen hinweg leiten.

Weitere Informationen finden Sie unter [Transit Gateways](#).

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die EC2 Amazon-Instance in unterschiedlichen VPCs Amazon-Regionen befinden

### Transit verwenden VPC

Als Alternative zur Verwendung von VPC Peering besteht eine weitere gängige Strategie zur Verbindung mehrerer geografisch verteilter VPCs und entfernter Netzwerke darin, einen Transit einzurichten VPC, der als globales Netzwerk-Transitzentrum dient. Ein Transit VPC vereinfacht die Netzwerkverwaltung und minimiert die Anzahl der Verbindungen, die für die Verbindung mehrerer VPCs Netzwerke und entfernter Netzwerke erforderlich sind. Dieses Design kann Zeit und Aufwand verringern und auch Kosten reduzieren, da es virtuell ohne die herkömmlichen Ausgaben implementiert wird, die beim Einrichten einer physischen Präsenz in einem Co-Location-Transit-Hub oder beim Bereitstellen physischer Netzwerkausstattung anfallen.



## Verbindungen zwischen VPCs verschiedenen Regionen

Sobald Transit Amazon eingerichtet VPC ist, kann eine Anwendung, die in einem „Spoke“ VPC in einer Region bereitgestellt wird, eine Verbindung zu einem ElastiCache Cache in einem „Spoke“ VPC in einer anderen Region herstellen.

Um auf einen Cache in einem anderen VPC innerhalb einer anderen AWS Region zuzugreifen

1. Stellen Sie eine VPC Transit-Lösung bereit. Weitere Informationen finden Sie unter [AWS - Transit-Gateway](#).
2. Aktualisieren Sie die VPC Routing-Tabellen in der App und im CacheVPCs, um den Datenverkehr über das VGW (Virtual Private Gateway) und die VPN Appliance weiterzuleiten. Im Fall von dynamischem Routing mit Border Gateway Protocol (BGP) werden Ihre Routen möglicherweise automatisch weitergegeben.
3. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen aus dem IP-Bereich der Anwendungsinstanz zuzulassen. Beachten Sie, dass Sie in diesem Szenario nicht auf die Sicherheitsgruppe des Anwendungsservers verwiesen können.

Beim regionsübergreifenden Zugriff auf einen Cache entstehen Netzwerklatenzen und fallen zusätzliche, regionsübergreifende Datenübertragungskosten an.

Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird

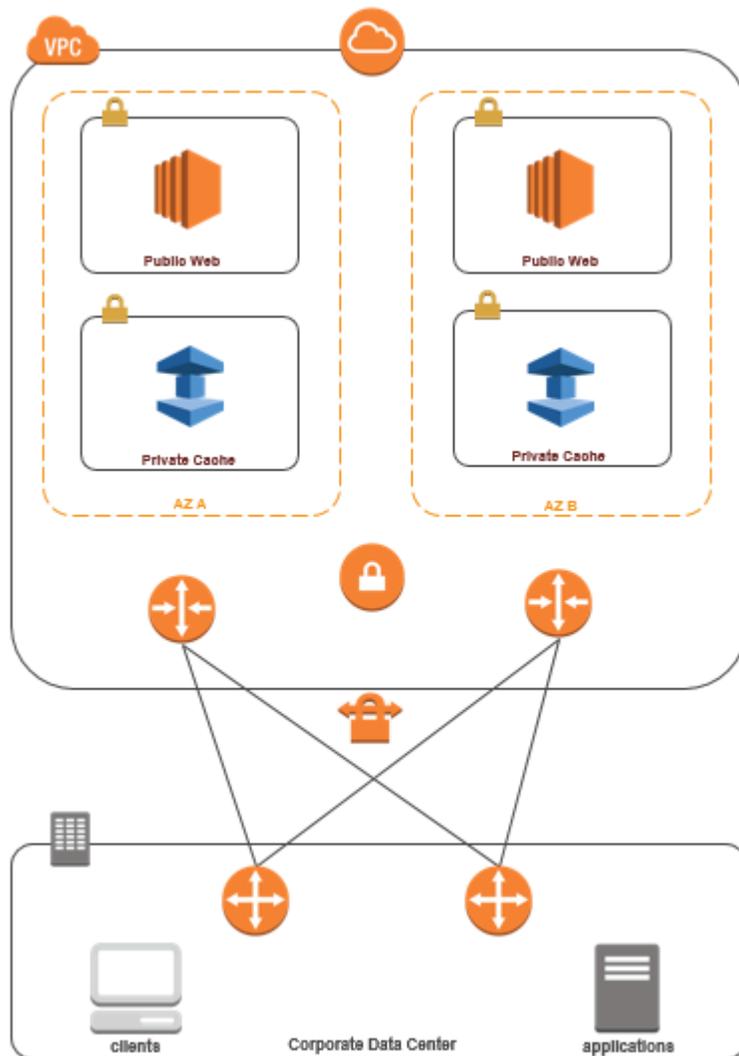
Ein anderes mögliches Szenario ist eine Hybridarchitektur, bei der Clients oder Anwendungen im Rechenzentrum des Kunden möglicherweise auf einen ElastiCache Cache im zugreifen müssen VPC. Dieses Szenario wird auch unterstützt, sofern eine Konnektivität zwischen den Kunden VPC und dem Rechenzentrum VPN entweder über Direct Connect besteht.

## Themen

- [Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mithilfe von VPN Konnektivität ausgeführt wird](#)
- [Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mit Direct Connect ausgeführt wird](#)

Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mithilfe von VPN Konnektivität ausgeführt wird

Das folgende Diagramm veranschaulicht den Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die in Ihrem Unternehmensnetzwerk ausgeführt wird, mithilfe von VPN Verbindungen.



Herstellen ElastiCache einer Verbindung zu Ihrem Rechenzentrum über ein VPN

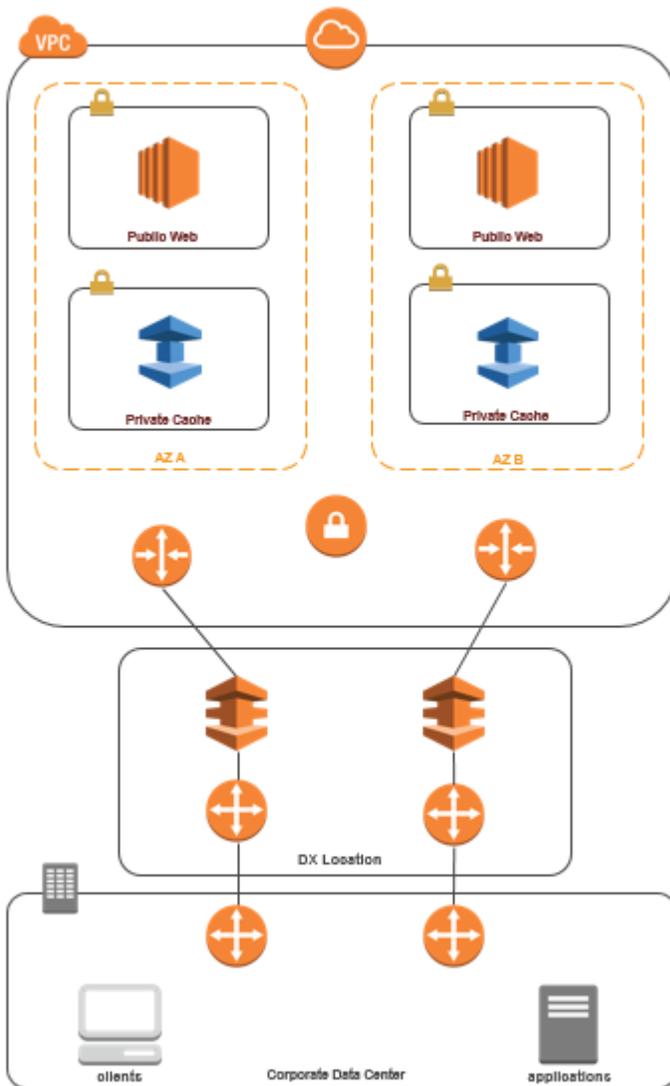
Um über eine Verbindung auf einen VPC Cache in einer lokalen Anwendung zuzugreifen VPN

1. Stellen Sie VPN Konnektivität her, indem Sie Ihrem ein Hardware-Virtual Private Gateway hinzufügen. VPC Weitere Informationen finden Sie unter [Hinzufügen eines Hardware-Virtual-Private-Gateways zu Ihrem VPC](#).
2. Aktualisieren Sie die VPC Routingtabelle für das Subnetz, in dem Ihr ElastiCache Cache bereitgestellt wird, um Datenverkehr von Ihrem lokalen Anwendungsserver zuzulassen. Im Fall von dynamischem Routing werden BGP Ihre Routen möglicherweise automatisch weitergegeben.
3. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen von den lokalen Anwendungsservern zuzulassen.

Der Zugriff auf einen Cache über eine VPN Verbindung führt zu Netzwerklatenzen und zusätzlichen Datenübertragungskosten.

Zugreifen auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mit Direct Connect ausgeführt wird

Das folgende Diagramm veranschaulicht den Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die in Ihrem Unternehmensnetzwerk mit Direct Connect ausgeführt wird.



Über Direct Connect eine Verbindung zu ElastiCache Ihrem Rechenzentrum herstellen

So greifen Sie mit Direct Connect von einer in Ihrem Netzwerk ausgeführten Anwendung auf einen ElastiCache Cache zu

1. Richten Sie Direct Connect-Konnektivität ein. Weitere Informationen finden Sie unter [Erste Schritte mit AWS Direct Connect](#).
2. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen von den lokalen Anwendungsservern zuzulassen.

Beim Zugriff auf einen Cache über eine DX-Verbindung können Netzwerklatenzen entstehen und zusätzliche Datenübertragungskosten anfallen.

## Eine virtuelle private Cloud erstellen (VPC)

In diesem Beispiel erstellen Sie ein Amazon VPC mit einem privaten Subnetz für jede Availability Zone.

### Amazon erstellen VPC (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die VPC Amazon-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im VPC Dashboard die Option Erstellen ausVPC.
3. Wählen Sie unter Zu erstellende Ressourcen die Option VPCund mehr aus.
4. Wählen Sie unter Anzahl der Availability Zones (AZs) die Anzahl der Availability Zones aus, in denen Sie Ihre Subnetze starten möchten.
5. Wählen Sie unter Anzahl der öffentlichen Subnetze die Anzahl der öffentlichen Subnetze aus, die Sie zu Ihren hinzufügen möchten. VPC
6. Wählen Sie unter Anzahl der privaten Subnetze die Anzahl der privaten Subnetze aus, die Sie zu Ihren hinzufügen möchten. VPC

#### Tip

Notieren Sie sich Ihre Subnetz-IDs und welches öffentlich und welches privat ist. Sie benötigen diese Informationen später, wenn Sie Ihre Cluster starten und Ihrem Amazon eine EC2 Amazon-Instance hinzufügenVPC.

7. Erstellen Sie eine VPC Amazon-Sicherheitsgruppe. Sie verwenden diese Gruppe für Ihren Cache-Cluster und Ihre EC2 Amazon-Instance.
  - a. Wählen Sie im Navigationsbereich der Amazon VPC Management Console die Option Sicherheitsgruppen aus.
  - b. Wählen Sie Sicherheitsgruppen erstellen aus.
  - c. Geben Sie in den entsprechenden Feldern einen Namen und eine Beschreibung für Ihre Sicherheitsgruppe ein. Wählen Sie in dem VPCFeld die Kennung für Ihren Amazon ausVPC.

**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 <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Destination <a href="#">Info</a>	Description - optional <a href="#">Info</a>
All traffic	All	All	Custom	

- d. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Ja, erstellen aus.
8. Definieren Sie eine Netzwerkeingangsregel für Ihre Sicherheitsgruppe. Diese Regel ermöglicht es Ihnen, mithilfe von Secure Shell (SSH) eine Verbindung zu Ihrer EC2 Amazon-Instance herzustellen.
    - a. Wählen Sie in der Navigationsliste Security Groups aus.
    - b. Suchen Sie Ihre Sicherheitsgruppe in der Liste und wählen Sie sie aus.
    - c. Wählen Sie unter Security Group die Registerkarte Inbound aus. Wählen Sie im Feld Neue Regel erstellen die Option SSH und anschließend Regel hinzufügen aus.
    - d. Legen Sie die folgenden Werte für Ihre neue Regel für eingehende Nachrichten fest, um den HTTP Zugriff zu ermöglichen:
      - Typ: HTTP
      - Quelle: 0.0.0.0/0

Wählen Sie Apply Rule Changes aus.

Jetzt sind Sie bereit, eine Cache-Subnetzgruppe zu erstellen und einen Cache-Cluster in Ihrem Amazon VPC zu starten.

- [Erstellen einer Subnetzgruppe](#)

- [Erstellen eines Memcached-Clusters \(Konsole\)](#).
- [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

## Verbindung zu einem Cache herstellen, der in einem Amazon läuft VPC

Dieses Beispiel zeigt, wie Sie eine EC2 Amazon-Instance in Ihrem Amazon startenVPC. Sie können sich dann bei dieser Instance anmelden und auf den ElastiCache Cache zugreifen, der im Amazon ausgeführt wirdVPC.

### Verbindung zu einem Cache herstellen, der in einem Amazon läuft VPC (Konsole)

In diesem Beispiel erstellen Sie eine EC2 Amazon-Instance in Ihrem AmazonVPC. Sie können diese EC2 Amazon-Instance verwenden, um eine Verbindung zu Cache-Knoten herzustellen, die im Amazon ausgeführt VPC werden.

#### Note

Informationen zur Nutzung von Amazon EC2 finden Sie im [Amazon-Leitfaden „EC2Erste Schritte“](#) in der [EC2Amazon-Dokumentation](#).

So erstellen Sie VPC mit der EC2 Amazon-Konsole eine EC2 Amazon-Instance in Ihrem Amazon

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie in der Konsole Launch Instance aus und folgen Sie diesen Schritten:
3. Wählen Sie auf der Seite Choose an Amazon Machine Image (AMI) die 64-Bit-Version von Amazon Linux AMI aus und wählen Sie dann Select aus.
4. Wählen Sie auf der Seite Instance-Typ auswählen die Option 3 aus. Instance konfigurieren.
5. Treffen Sie auf der Seite Configure Instance Details (Instance-Details konfigurieren) folgende Auswahl:
  - a. Wählen Sie in der Netzwerkliste Ihr Amazon ausVPC.
  - b. Wählen Sie in der Liste Subnet (Subnetz) Ihr öffentliches Subnetz aus.

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

Purchasing option  Request Spot Instances

Network

Subnet    
250 IP Addresses available

Public IP  Automatically assign a public IP address to your instances

Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie 4. Speicher hinzufügen.

6. Wählen Sie auf der Seite Add Storage (Speicher hinzufügen) die Option 5. Taggen der Instance.
7. Geben Sie auf der Seite Tag Instance einen Namen für Ihre EC2 Amazon-Instance ein und wählen Sie dann 6. Konfigurieren der Sicherheitsgruppe.
8. Wählen Sie auf der Seite Configure Security Group (Sicherheitsgruppe konfigurieren) die Option Select an existing security group (Bestehende Sicherheitsgruppe auswählen) aus. Weitere Informationen zu Sicherheitsgruppen finden Sie unter [EC2Amazon-Sicherheitsgruppen für Linux-Instances](#).

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group:  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description
<input type="checkbox"/> sg-1a3d2178	default	default VPC security group
<input checked="" type="checkbox"/> sg-f13d2193	my-vpc-security-group	Testing

Wählen Sie den Namen Ihrer VPC Amazon-Sicherheitsgruppe und klicken Sie dann auf Überprüfen und starten.

9. Wählen Sie auf der Seite Review Instance and Launch (Instance prüfen und starten) die Option Launch (Starten) aus.

10. Geben Sie im Fenster Select an existing key pair or create a new key pair (Ein vorhandenes Schlüsselpaar auswählen oder ein neues Schlüsselpaar erstellen) ein Schlüsselpaar an, das Sie mit dieser Instance verwenden möchten.

 Note

Informationen zur Verwaltung von Schlüsselpaaren finden Sie im [Amazon EC2 Getting Started Guide](#).

11. Wenn Sie bereit sind, Ihre EC2 Amazon-Instance zu starten, wählen Sie Launch.

Sie können der EC2 Amazon-Instance, die Sie gerade erstellt haben, jetzt eine Elastic IP-Adresse zuweisen. Sie müssen diese IP-Adresse verwenden, um eine Verbindung zur EC2 Amazon-Instance herzustellen.

So weisen Sie eine Elastic IP-Adresse zu (Konsole)

1. Öffnen Sie die VPC Amazon-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie in der Navigationsliste Elastic aus/IPv4s.
3. Wählen Sie Elastic-IP-Adresse zuweisen aus.
4. Übernehmen Sie im Dialogfeld Allocate Elastic IP address (Elastic IP-Adresse zuweisen) die standardmäßige Network Border Group (Netzwerkrenzgruppe) und klicken Sie auf Allocate (Zuweisen).
5. Wählen Sie die Elastic IP-Adresse aus, die Sie gerade aus der Liste zugewiesen haben, und klicken Sie dann auf Associate Address (Adresse verknüpfen).
6. Wählen Sie im Dialogfeld Associate Address im Feld Instance die ID der EC2 Amazon-Instance aus, die Sie gestartet haben.

Markieren Sie im Feld Private IP address (Private IP-Adresse) das Kästchen, um die private IP-Adresse abzurufen, und wählen Sie dann Associate (Verknüpfen).

Sie können SSH jetzt über die von Ihnen erstellte Elastic IP-Adresse eine Verbindung zur EC2 Amazon-Instance herstellen.

## So stellen Sie eine Verbindung zu Ihrer EC2 Amazon-Instance her

- Öffnen Sie ein Befehlsfenster. Geben Sie an der Eingabeaufforderung den folgenden Befehl aus, wobei Sie `mykeypair.pem` durch den Namen Ihrer Schlüsselpaardatei und `54.207.55.251` durch Ihre Elastic IP-Adresse ersetzen.

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```

### Important

Melden Sie sich noch nicht von Ihrer EC2 Amazon-Instance ab.

Sie sind jetzt bereit, mit Ihrem ElastiCache Cluster zu interagieren. Vorher aber müssen Sie, wenn Sie es noch nicht getan haben, das Dienstprogramm `telnet` installieren.

## Installieren von `telnet` und Interaktion mit Ihrem Cache-Cluster (AWS CLI)

- Öffnen Sie ein Befehlsfenster. Führen Sie an der Eingabeaufforderung den folgenden Befehl aus. Geben Sie bei der Eingabeaufforderung zur Bestätigung `y` ein.

```
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!
```

Sie können jetzt entweder VPC mit Memcached oder Redis eine Verbindung zu einem herstellen.

Mit Memcached eine Verbindung zu einem herstellen VPC

1. Rufen Sie die ElastiCache Konsole unter auf <https://console.aws.amazon.com/elasticache/> und rufen Sie den Endpunkt für einen der Knoten in Ihrem Cache-Cluster ab. Weitere Informationen finden Sie unter [Finden von Verbindungsendpunkten](#).
2. Verwenden Sie telnet, um über Port 11211 eine Verbindung zu Ihrem Cache-Knotenendpunkt herzustellen. Ersetzen Sie den unten gezeigten Hostnamen durch den Hostnamen Ihres Cache-Knotens.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

Sie sind nun mit der Cache-Engine verbunden und können Befehle ausgeben. In diesem Beispiel fügen Sie dem Cache ein Datenelement hinzu und rufen es dann sofort danach ab. Schließlich trennen Sie die Verbindung zum Cache-Knoten.

Um einen Schlüssel und einen Wert zu speichern, geben Sie die folgenden zwei Zeilen ein:

```
add mykey 0 3600 28
This is the value for mykey
```

Die Cache-Engine antwortet wie folgt:

```
OK
```

So rufen Sie den Wert für mykey Geben Sie für den Wert Folgendes ein:

```
get mykey
```

Die Cache-Engine antwortet wie folgt:

```
VALUE mykey 0 28
This is the value for my key
END
```

Geben Sie Folgendes ein, um die Verbindung zum Cache-Modul zu trennen:

```
quit
```

Verbindung zu einem VPC mit Redis herstellen

1. Rufen Sie die ElastiCache Konsole unter auf <https://console.aws.amazon.com/elasticache/> und rufen Sie den Endpunkt für einen der Knoten in Ihrem Cache-Cluster ab. Weitere Informationen [finden Sie unter Verbindungsendpunkte](#) für Redis suchen.
2. Verwenden Sie telnet, um über Port 6379 eine Verbindung zu Ihrem Cache-Knotenendpunkt herzustellen. Ersetzen Sie den unten gezeigten Hostnamen durch den Hostnamen Ihres Cache-Knotens.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Sie sind nun mit der Cache-Engine verbunden und können Befehle ausgeben. In diesem Beispiel fügen Sie dem Cache ein Datenelement hinzu und rufen es dann sofort danach ab. Schließlich trennen Sie die Verbindung zum Cache-Knoten.

Um einen Schlüssel und einen Wert zu speichern, geben Sie die folgenden zwei Zeilen ein:

```
set mykey myvalue
```

Die Cache-Engine antwortet wie folgt:

```
OK
```

So rufen Sie den Wert für mykey Geben Sie für den Wert Folgendes ein:

```
get mykey
```

Geben Sie Folgendes ein, um die Verbindung zum Cache-Modul zu trennen:

```
quit
```

3. Rufen Sie die ElastiCache Konsole unter auf <https://console.aws.amazon.com/elasticache/> und rufen Sie den Endpunkt für einen der Knoten in Ihrem Cache-Cluster ab. Weitere Informationen [finden Sie unter Verbindungsendpunkte](#) für OSS Redis suchen.

4. Verwenden Sie telnet, um über Port 6379 eine Verbindung zu Ihrem Cache-Knotenendpunkt herzustellen. Ersetzen Sie den unten gezeigten Hostnamen durch den Hostnamen Ihres Cache-Knotens.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 6379
```

Sie sind nun mit der Cache-Engine verbunden und können Befehle ausgeben. In diesem Beispiel fügen Sie dem Cache ein Datenelement hinzu und rufen es dann sofort danach ab. Schließlich trennen Sie die Verbindung zum Cache-Knoten.

Geben Sie zum Speichern eines Schlüssels und eines Werts Folgendes ein:

```
set mykey myvalue
```

Die Cache-Engine antwortet wie folgt:

```
OK
```

So rufen Sie den Wert für *mykey* ab. Geben Sie für den Wert Folgendes ein:

```
get mykey
```

Die Cache-Engine antwortet wie folgt:

```
get mykey
myvalue
```

Geben Sie Folgendes ein, um die Verbindung zum Cache-Modul zu trennen:

```
quit
```

### Important

Um zusätzliche Gebühren für Ihr AWS Konto zu vermeiden, sollten Sie alle AWS Ressourcen löschen, die Sie nicht mehr benötigen, nachdem Sie diese Beispiele ausprobiert haben.

## ElastiCache API und VPC Schnittstellenendpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihren Endpunkten VPC und ElastiCache API Amazon-Endpunkten herstellen, indem Sie einen VPC Schnittstellenendpunkt erstellen. Schnittstellen-Endpunkte werden mit Strom versorgt von [AWS PrivateLink](#). AWS PrivateLink ermöglicht Ihnen den privaten Zugriff auf ElastiCache API Amazon-Operationen ohne Internet-Gateway, NAT Gerät, VPN Verbindung oder AWS Direct Connect-Verbindung.

Instances in Ihrem VPC System benötigen keine öffentlichen IP-Adressen, um mit ElastiCache API Amazon-Endpunkten zu kommunizieren. Ihre Instances benötigen außerdem keine öffentlichen IP-Adressen, um die verfügbaren ElastiCache API Operationen nutzen zu können. Der Verkehr zwischen Ihnen VPC und Amazon verlässt das Amazon-Netzwerk ElastiCache nicht. Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic Network-Schnittstellen in Ihren Subnetzen dargestellt. Weitere Informationen zu Elastic Network Interfaces finden Sie unter [Elastic Network Interfaces](#) im EC2 Amazon-Benutzerhandbuch.

- Weitere Informationen zu VPC Endpunkten finden Sie unter [Interface VPC endpoints \(AWS PrivateLink\)](#) im VPC Amazon-Benutzerhandbuch.
- [Weitere Informationen zu ElastiCache API Vorgängen finden Sie unter ElastiCache API Operationen.](#)

Wenn Sie nach dem Erstellen eines VPC Schnittstellenendpunkts [private DNS](#) Hostnamen für den Endpunkt aktivieren, wird dies der ElastiCache Standardendpunkt sein (<https://elasticache.Region.amazonaws.com>) wird zu Ihrem Endpunkt aufgelöst. VPC Wenn Sie private DNS Hostnamen nicht aktivieren, VPC stellt Amazon einen DNS Endpunktnamen bereit, den Sie im folgenden Format verwenden können:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Weitere Informationen finden Sie unter [Interface VPC Endpoints \(AWS PrivateLink\)](#) im VPC Amazon-Benutzerhandbuch. ElastiCache unterstützt das Aufrufen all seiner [API Aktionen](#) in Ihrem VPC.

### Note

Private DNS Hostnamen können nur für einen VPC Endpunkt in der VPC aktiviert werden. Wenn Sie einen zusätzlichen VPC Endpunkt erstellen möchten, sollte der private DNS Hostname dafür deaktiviert werden.

## Überlegungen zu Endpunkten VPC

Bevor Sie einen VPC ElastiCache API Schnittstellenendpunkt für Amazon-Endgeräte einrichten, sollten Sie die [Eigenschaften und Einschränkungen der Schnittstellenendpunkte](#) im VPCAmazon-Benutzerhandbuch lesen. Alle ElastiCache API Vorgänge, die für die Verwaltung von ElastiCache Amazon-Ressourcen relevant sind, stehen Ihnen zur VPC Verfügung AWS PrivateLink.

VPCEndpunktrichtlinien werden für ElastiCache API Endgeräte unterstützt. Standardmäßig ist der vollständige Zugriff auf ElastiCache API Vorgänge über den Endpunkt zulässig. Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Dienste mit VPC Endpunkten](#) im VPCAmazon-Benutzerhandbuch.

## Erstellen eines VPC Schnittstellen-Endpunkts für ElastiCache API

Sie können einen VPC Endpunkt für Amazon entweder ElastiCache API mit der VPC Amazon-Konsole oder mit dem erstellen AWS CLI. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im VPCAmazon-Benutzerhandbuch.

Nachdem Sie einen VPC Schnittstellenendpunkt erstellt haben, können Sie private DNS Hostnamen für den Endpunkt aktivieren. Wenn Sie dies tun, der ElastiCache Standard-Amazon-Endpunkt (<https://elasticache.Region.amazonaws.com>) wird zu Ihrem Endpunkt aufgelöst. VPC Für die AWS Regionen China (Peking) und China (Ningxia) können Sie API Anfragen mit dem VPC Endpunkt stellen, indem Sie die Option für Peking und `elasticache.cn-north-1.amazonaws.com.cn` für Ningxia verwenden. `elasticache.cn-northwest-1.amazonaws.com.cn` Weitere Informationen finden Sie unter [Zugreifen auf einen Service über einen Schnittstellenendpunkt](#) im VPCAmazon-Benutzerhandbuch.

## Erstellen einer VPC Endpunktrichtlinie für Amazon ElastiCache API

Sie können Ihrem VPC Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf den steuert ElastiCache API. Die Richtlinie legt Folgendes fest:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Dienste mit VPC Endpunkten](#) im VPCAmazon-Benutzerhandbuch.

## Example VPC Endpunktrichtlinie für ElastiCache API Aktionen mit Valkey oder Redis OSS

Das Folgende ist ein Beispiel für eine Endpunktrichtlinie für die ElastiCache API. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten ElastiCache API Aktionen.

```
{
 "Statement": [{
 "Principal": "*",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:ModifyCacheCluster",
 "elasticache:CreateSnapshot"
],
 "Resource": "*"
 }]
}
```

## Example VPC Endpunkt-Richtlinie für ElastiCache (Memcached-) Aktionen API

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für die ElastiCache API. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten ElastiCache API Aktionen.

```
{
 "Statement": [{
 "Principal": "*",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:ModifyCacheCluster"
],
 "Resource": "*"
 }]
}
```

## Example VPC Endpunktrichtlinie, die jeglichen Zugriff von einem bestimmten Konto aus verweigert AWS

Die folgende VPC Endpunktrichtlinie verweigert AWS das Konto **123456789012** der gesamte Zugriff auf Ressourcen über den Endpunkt. Die Richtlinie erlaubt alle Aktionen von anderen Konten.

```
{
 "Statement": [{
 "Action": "*",
 "Effect": "Allow",
 "Resource": "*",
 "Principal": "*"
 },
 {
 "Action": "*",
 "Effect": "Deny",
 "Resource": "*",
 "Principal": {
 "AWS": [
 "123456789012"
]
 }
 }
]
}
```

## Subnetze und Subnetzgruppen

Eine Subnetzgruppe ist eine Sammlung von Subnetzen (normalerweise privat), die Sie für Ihre selbst entworfenen Cluster festlegen können, die in einer Amazon Virtual Private Cloud (VPC) -Umgebung ausgeführt werden.

Wenn Sie einen selbst entworfenen Cluster in einem Amazon erstellen VPC, müssen Sie eine Subnetzgruppe verwenden. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen.

ElastiCache stellt eine IPv4 Standard-Subnetzgruppe bereit, oder Sie können wählen, ob Sie eine neue erstellen möchten. Wenn Sie IPv6 wählen, müssen Sie eine Subnetzgruppe mit einem IPv6 CIDR Block erstellen. Wenn Sie sich für Dual-Stack entscheiden, müssen Sie anschließend einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

ElastiCache Serverless verwendet keine Subnetzgruppenressource, sondern verwendet stattdessen direkt bei der Erstellung eine Liste von Subnetzen.

In diesem Abschnitt wird beschrieben, wie Sie Subnetze und Subnetzgruppen erstellen und nutzen, um den Zugriff auf Ihre Ressourcen zu verwalten. ElastiCache

Weitere Informationen zur Nutzung von Subnetzgruppen in einer VPC Amazon-Umgebung finden Sie unter [Zugreifen auf Ihren ElastiCache Cluster oder Ihre Replikationsgruppe](#).

## Themen

- [Erstellen einer Subnetzgruppe](#)
- [Zuweisen einer Subnetzgruppe zu einem Cache](#)
- [Ändern einer Subnetzgruppe](#)
- [Löschen einer Subnetzgruppe](#)

## Erstellen einer Subnetzgruppe

Eine Cache-Subnetzgruppe ist eine Sammlung von Subnetzen, die Sie möglicherweise für Ihre Caches in einem festlegen möchten. VPC Wenn Sie einen Cache in einem startenVPC, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache Verwendet dann diese Cache-Subnetzgruppe, um jedem Cache-Knoten im Cache IP-Adressen innerhalb dieses Subnetzes zuzuweisen.

Wenn Sie eine neue Subnetzgruppe erstellen, notieren Sie sich die Anzahl der verfügbaren IP-Adressen. Wenn das Subnetz nur über wenige freie IP-Adressen verfügt, beschränkt dies auch die Anzahl der neuen Knoten, die Sie einem Cluster hinzufügen können. Um dieses Problem zu lösen, können Sie einer Subnetzgruppe weitere Subnetze zuweisen, um ausreichend IP-Adressen in der Availability Zone Ihres Clusters bereitzustellen. Danach können Sie dem Cluster weitere Knoten hinzufügen.

Wenn Sie IPV4 als Netzwerktyp wählen, ist eine Standard-Subnetzgruppe verfügbar, oder Sie können wählen, ob Sie eine neue erstellen möchten. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen, die Ihren Knoten zugeordnet werden sollen. Wenn Sie Dual-Stack oder wählen, werden Sie angewiesenIPV6, Dual-Stack oder Subnetze zu erstellen. IPV6 Weitere Informationen zu Netzwerktypen finden Sie unter [Netzwerktyp](#). Weitere Informationen finden Sie unter [Erstellen Sie ein Subnetz](#) in Ihrem. VPC

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Subnetzgruppe mit den Namen mysubnetgroup (Konsole) AWS CLI, die und die erstellen. ElastiCache API

### Erstellen einer Subnetzgruppe (Konsole)

Im folgenden Verfahren wird das Erstellen einer Subnetzgruppe (Konsole) erläutert.

### Erstellen einer DB-Sicherheitsgruppe (Konsole)

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie in der Navigationsliste auf Subnetzgruppen.
3. Wählen Sie Create subnet group (Subnetz-Grupp erstellen) aus.
4. Gehen Sie im Assistenten Subnetzgruppe erstellen wie folgt vor. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Erstellen aus.
  - a. Geben Sie im Feld Name einen Namen für Ihre Subnetzgruppe ein.
  - b. Geben Sie im Feld Description eine Beschreibung für Ihre Subnetzgruppe ein.

- c. Wählen Sie im VPCID-Feld Ihr Amazon ausVPC.
  - d. Standardmäßig werden alle Subnetze ausgewählt. Klicken Sie im Bereich Ausgewählte Subnetze auf Verwalten und wählen Sie die Availability Zones oder [Local Zones](#) sowie Ihre privaten Subnetze IDs aus, und wählen Sie dann Wählen aus.
5. Klicken Sie in der angezeigten Bestätigungsmeldung auf Close.

Ihre neue Subnetzgruppe wird in der Liste der Subnetzgruppen der Konsole angezeigt. ElastiCache Unten im Fenster können Sie die Subnetzgruppe auswählen, um Details wie die der Gruppe zugeordneten Subnetze anzuzeigen.

### Erstellen einer Subnetzgruppe (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `create-cache-subnet-group` ein, um eine Subnetzgruppe zu erstellen.

Für Linux, macOS oder Unix:

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

Für Windows:

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

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

```
{
 "CacheSubnetGroup": {
 "VpcId": "vpc-37c3cd17",
 "CacheSubnetGroupDescription": "Testing",
 "Subnets": [
 {
 "SubnetIdentifier": "subnet-53df9c3a",
 "SubnetAvailabilityZone": {
```

```
 "Name": "us-west-2a"
 }
],
 "CacheSubnetGroupName": "mysubnetgroup"
 }
}
```

Weitere Informationen finden Sie im Thema AWS CLI [create-cache-subnet-group](#).

## Zuweisen einer Subnetzgruppe zu einem Cache

Nachdem Sie eine Subnetzgruppe erstellt haben, können Sie einen Cache in einem Amazon VPC starten. Weitere Informationen finden Sie unter den folgenden Topics.

- Memcached-Cluster – Informationen zum Starten eines Memcached-Clusters finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#). Wählen Sie in Schritt 7.a (Erweiterte Memcached-Einstellungen) eine Subnetzgruppe aus. VPC
- Eigenständiger Valkey- oder OSS Redis-Cluster — Informationen zum Starten eines Valkey- oder Redis-Clusters mit einem Knoten finden Sie unter. OSS [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#) Wählen Sie in Schritt 7.a (Erweiterte OSS Redis-Einstellungen) eine Subnetzgruppe aus. VPC
- Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) — Informationen zum Starten einer Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus deaktiviert) in einer finden Sie unter. VPC [Eine Valkey- oder Redis-Replikationsgruppe OSS \(Clustermodus deaktiviert\) von Grund auf neu erstellen](#) Wählen Sie in Schritt 7.b (Erweiterte OSS Redis-Einstellungen) eine Subnetzgruppe aus. VPC
- Valkey- oder Redis-Replikationsgruppe OSS (Clustermodus aktiviert) —. [Erstellen eines Valkey- oder Redis-Clusters OSS \(Clustermodus aktiviert\) \(Konsole\)](#) Wählen Sie in Schritt 6.i (Erweiterte OSS Redis-Einstellungen) eine Subnetzgruppe aus. VPC

## Ändern einer Subnetzgruppe

Sie können die Beschreibung einer Subnetzgruppe oder die Liste der Subnetze ändern, die der Subnetzgruppe IDs zugeordnet sind. Es ist nicht möglich, Subnetz-IDs aus einer Subnetzgruppe zu löschen, wenn das Subnetz derzeit von einem Cache verwendet wird.

Das folgende Verfahren zeigt, wie Sie eine Subnetzgruppe ändern.

### Subnetzgruppen ändern (Konsole)

So ändern Sie eine Subnetzgruppe

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die ElastiCache Konsole unter. <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
3. Wählen Sie in der Liste der Subnetzgruppen das Optionsfeld für die gewünschte Subnetzgruppe aus und klicken Sie auf Ändern.
4. Wählen Sie im Bereich Ausgewählte Subnetze die Option Verwalten aus.
5. Nehmen Sie die gewünschten Änderungen an den ausgewählten Subnetzen vor und klicken Sie auf Auswählen.
6. Klicken Sie auf Änderungen speichern, um Ihre Änderungen zu speichern.

### Ändern von Subnetzgruppen (AWS CLI)

Geben Sie in einem Befehlszeilenfenster den Befehl `modify-cache-subnet-group` ein, um eine Subnetzgruppe zu ändern.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-subnet-group \
 --cache-subnet-group-name mysubnetgroup \
 --cache-subnet-group-description "New description" \
 --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Für Windows:

```
aws elasticache modify-cache-subnet-group ^
 --cache-subnet-group-name mysubnetgroup ^
 --cache-subnet-group-description "New description" ^
```

```
--subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Die Ausgabe dieses Befehls sieht ähnlich wie folgt aus:

```
{
 "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"
 }
}
```

Weitere Informationen finden Sie im AWS CLI Thema [modify-cache-subnet-group](#).

## Löschen einer Subnetzgruppe

Wenn Sie eine Subnetzgruppe nicht mehr benötigen, können Sie sie löschen. Sie können eine Subnetzgruppe, die derzeit von einem Cache verwendet wird, nicht löschen.

Das folgende Verfahren zeigt, wie Sie eine Subnetzgruppe löschen.

### Löschen einer Subnetzgruppe (Konsole)

So löschen Sie eine Subnetzgruppe

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Subnetzgruppe aus.
3. Wählen Sie in der Liste der Subnetzgruppen die zu löschende Subnetzgruppe aus und klicken Sie auf Delete.
4. Wenn Sie aufgefordert werden, diesen Vorgang zu bestätigen, geben Sie den Namen der Subnetzgruppe in das Texteingabefeld ein und wählen Sie Löschen aus.

### Löschen einer Subnetzgruppe (AWS CLI)

Rufen Sie mit dem AWS CLI den Befehl `delete-cache-subnet-group` mit dem folgenden Parameter auf:

- `--cache-subnet-group-name mysubnetgroup`

Für Linux, macOS oder Unix:

```
aws elasticache delete-cache-subnet-group \
 --cache-subnet-group-name mysubnetgroup
```

Für Windows:

```
aws elasticache delete-cache-subnet-group ^
 --cache-subnet-group-name mysubnetgroup
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie im AWS CLI Thema [delete-cache-subnet-group](#).

# Identity and Access Management für Amazon ElastiCache

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um ElastiCache Ressourcen zu verwenden. IAM ist eine AWS-Service, die Sie ohne zusätzliche Kosten verwenden können.

## Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So ElastiCache arbeitet Amazon mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)
- [Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff](#)
- [Zugriffskontrolle](#)
- [Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen](#)

## Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie arbeiten ElastiCache.

**Dienstbenutzer** — Wenn Sie den ElastiCache Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr ElastiCache Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie nicht auf eine Funktion zugreifen können ElastiCache, finden Sie weitere Informationen unter [Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff](#).

**Serviceadministrator** — Wenn Sie in Ihrem Unternehmen für die ElastiCache Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf ElastiCache. Es ist Ihre Aufgabe, zu bestimmen, auf welche ElastiCache Funktionen und Ressourcen Ihre Servicebenutzer

zugreifen sollen. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die Berechtigungen Ihrer Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehen IAM. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit verwenden kann ElastiCache, finden Sie unter [So ElastiCache arbeitet Amazon mit IAM](#).

IAM Administrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff darauf zu verwalten ElastiCache. Beispiele für ElastiCache identitätsbasierte Richtlinien, die Sie in verwenden können IAM, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

## Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM Benutzer authentifizieren (angemeldet bei AWS) oder indem Sie eine IAM Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center-) Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie AWS mithilfe eines Verbunds darauf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportale anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit der Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM Benutzerhandbuch unter AWS API Anfragen signieren](#).

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center

Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) AWS im IAM Benutzerhandbuch](#).

## AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter [Aufgaben, für die Root-Benutzeranmeldedaten erforderlich](#) sind. IAM

## Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center Benutzerhandbuch.

## IAM-Benutzer und -Gruppen

Ein [IAMBenutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch

spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind](#).

Eine [IAMGruppe](#) ist eine Identität, die eine Sammlung von IAM Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Wann sollte ein IAM Benutzer \(statt einer Rolle\) erstellt werden?](#) im IAMBenutzerhandbuch.

## IAMRollen

Eine [IAMRolle](#) ist eine Identität innerhalb von Ihrem AWS-Konto, für die bestimmte Berechtigungen gelten. Sie ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Sie können vorübergehend eine IAM Rolle in der übernehmen, AWS Management Console indem Sie die [Rollen wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI AWS API OR-Operation aufrufen oder eine benutzerdefinierte Operation verwendenURL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie unter [Methoden zur Übernahme einer Rolle](#) im IAMBenutzerhandbuch.

IAMRollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle für einen externen Identitätsanbieter](#). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Temporäre IAM Benutzerberechtigungen — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- Kontoübergreifender Zugriff — Sie können eine IAM Rolle verwenden, um einer Person (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAM Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen. AWS-Services Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Zugriffssitzungen weiterleiten (FAS) — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Berechtigungen des Prinzipals, der an aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. FASANfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
  - Servicerolle — Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine](#).
  - Dienstbezogene Rolle — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance

ausgeführt werden und AWS API Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt](#) werden.

Informationen darüber, ob Sie IAM Rollen oder IAM Benutzer verwenden sollten, finden [Sie im Benutzerhandbuch unter Wann sollte eine IAM Rolle \(anstelle eines IAM Benutzers\) erstellt](#) werden.

## Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS Form von JSON Dokumenten gespeichert. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter [Überblick über JSON Richtlinien](#).

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAMRichtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen aus dem AWS Management Console AWS CLI, dem oder dem abrufen AWS API.

## Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAM Richtlinien erstellen im Benutzerhandbuch](#). IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können. AWS-Konto Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie wählen können, finden Sie im IAM Benutzerhandbuch unter [Auswahl zwischen verwalteten Richtlinien und Inline-Richtlinien](#).

## Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien nicht IAM in einer ressourcenbasierten Richtlinie verwenden.

## Zugriffskontrolllisten (ACLs)

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3 und AWS WAF Amazon VPC sind Beispiele für Dienste, die Unterstützung bieten ACLs. Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAMBenutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAMBenutzerhandbuch unter [Berechtigungsgrenzen für IAM Entitäten](#).
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Geräte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter [Richtlinien zur Servicesteuerung](#) im AWS Organizations Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Sitzungsrichtlinien](#).

## Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt

wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAMBenutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

## So ElastiCache arbeitet Amazon mit IAM

Informieren Sie sich vor der Verwendung IAM zur Verwaltung des Zugriffs auf ElastiCache, welche IAM Funktionen zur Nutzung verfügbar sind ElastiCache.

IAMFunktionen, die Sie mit Amazon verwenden können ElastiCache

IAMFunktion	ElastiCache Unterstützung
<a href="#">Identitätsbasierte Richtlinien</a>	Ja
<a href="#">Ressourcenbasierte Richtlinien</a>	Nein
<a href="#">Richtlinienaktionen</a>	Ja
<a href="#">Richtlinienressourcen</a>	Ja
<a href="#">Bedingungsschlüssel für die Richtlinie</a>	Ja
<a href="#">ACLs</a>	Ja
<a href="#">ABAC(Tags in Richtlinien)</a>	Ja
<a href="#">Temporäre Anmeldeinformationen</a>	Ja
<a href="#">Hauptberechtigungen</a>	Ja
<a href="#">Servicerollen</a>	Ja
<a href="#">Service-verknüpfte Rollen</a>	Ja

Einen allgemeinen Überblick darüber, wie ElastiCache und wie andere AWS Dienste mit den meisten IAM Funktionen funktionieren, finden Sie IAM im IAMBenutzerhandbuch unter [AWS Dienste, die mit funktionieren](#).

## Identitätsbasierte Richtlinien für ElastiCache

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAM Richtlinien erstellen im Benutzerhandbuch](#). IAM

Mit IAM identitätsbasierten Richtlinien können Sie zulässige oder verweigernde Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie im IAM Benutzerhandbuch unter [Referenz zu IAM JSON Richtlinienelementen](#).

Beispiele für identitätsbasierte Richtlinien für ElastiCache

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

## Ressourcenbasierte Richtlinien in ElastiCache

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um den kontoübergreifenden Zugriff zu ermöglichen, können Sie in einer ressourcenbasierten Richtlinie ein ganzes Konto oder IAM Entitäten in einem anderen Konto als Prinzipal angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource

unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalidentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource gewähren. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie [IAMim IAMBenutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#).

## Politische Maßnahmen für ElastiCache

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur eine Genehmigung erforderlich ist und für die es keinen entsprechenden Vorgang gibt. API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der ElastiCache Aktionen finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#) in der Service Authorization Reference.

Bei den in der Richtlinie ElastiCache verwendeten Aktionen wird vor der Aktion das folgende Präfix verwendet:

```
elasticache
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
 "elasticache:action1",
 "elasticache:action2"
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "elasticache:Describe*"
```

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

## Politische Ressourcen für ElastiCache

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem [Amazon-Ressourcennamen \(ARN\)](#) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (\*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der ElastiCache Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

## Richtlinien-Bedingungsschlüssel für ElastiCache

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM Benutzer nur dann Zugriff auf eine Ressource gewähren, wenn sie mit seinem IAM Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinienelemente: Variablen und Tags](#).

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontext-Schlüssel für AWS globale Bedingungen](#) im IAMBenutzerhandbuch.

Eine Liste der ElastiCache Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

## Zugriffskontrolllisten (ACLs) in ElastiCache

UnterstütztACLs: Ja

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

## Attributbasierte Zugriffskontrolle ( ) mit ABAC ElastiCache

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen definiert. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABAC ist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu finden Sie ABAC unter [Was ist? ABAC](#) im IAM Benutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter [Verwenden der attributbasierten Zugriffskontrolle \(ABAC\)](#). IAM

## Temporäre Anmeldeinformationen verwenden mit ElastiCache

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen darüber, AWS-Services wie Sie mit temporären Anmeldeinformationen [arbeiten können AWS-Services](#), finden Sie IAM im IAM Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Kennwort anmelden. Wenn

Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAMBenutzerhandbuch.

Mit dem AWS CLI oder können Sie manuell temporäre Anmeldeinformationen erstellen AWS API. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen unter IAM](#).

## Serviceübergreifende Prinzipal-Berechtigungen für ElastiCache

Unterstützt Forward-Access-Sitzungen (FAS): Ja

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. FASANfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

## Servicerollen für ElastiCache

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine](#).

### Warning

Das Ändern der Berechtigungen für eine Servicerolle kann zu ElastiCache Funktionseinschränkungen führen. Bearbeiten Sie Servicerollen nur, ElastiCache wenn Sie dazu eine Anleitung erhalten.

## Dienstbezogene Rollen für ElastiCache

Unterstützt dienstbezogene Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer Service-Verknüpfung ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen finden Sie unter [AWS Dienste, die mit funktionieren](#). IAM Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

## Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Ressourcen zu erstellen oder zu ändern ElastiCache. Sie können auch keine Aufgaben mithilfe von AWS Management Console, AWS Command Line Interface (AWS CLI) oder ausführen AWS API. Um Benutzern die Berechtigung zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

Informationen zum Erstellen einer IAM identitätsbasierten Richtlinie anhand dieser JSON Beispieldokumente finden Sie unter [IAM Richtlinien erstellen](#) im IAM Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden ElastiCache, einschließlich des Formats von ARNs für jeden der Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der ElastiCache Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

## Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand ElastiCache Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie AWS im IAMBenutzerhandbuch unter [AWS Verwaltete Richtlinien oder Verwaltete Richtlinien für Jobfunktionen](#).
- Berechtigungen mit den geringsten Rechten anwenden — Wenn Sie Berechtigungen mit IAM Richtlinien festlegen, gewähren Sie nur die Berechtigungen, die für die Ausführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung IAM zum Anwenden von Berechtigungen finden Sie [IAMim Benutzerhandbuch unter Richtlinien und Berechtigungen](#). IAM
- Verwenden Sie Bedingungen in IAM Richtlinien, um den Zugriff weiter einzuschränken — Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen einzuschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um anzugeben, dass alle Anfragen mit gesendet werden müssenSSL. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese über einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter [IAMJSONRichtlinienelemente: Bedingung](#) im IAMBenutzerhandbuch.
- Verwenden Sie IAM Access Analyzer, um Ihre IAM Richtlinien zu validieren, um sichere und funktionale Berechtigungen zu gewährleisten. IAM Access Analyzer validiert neue und bestehende Richtlinien, sodass die Richtlinien der IAM Richtlinienensprache (JSON) und den IAM bewährten Methoden entsprechen. IAMAccess Analyzer bietet mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen, um Sie bei der Erstellung sicherer und funktionaler Richtlinien zu unterstützen. Weitere Informationen finden Sie unter [IAMAccess Analyzer-Richtlinienvvalidierung](#) im IAMBenutzerhandbuch.

- Multi-Faktor-Authentifizierung erforderlich (MFA) — Wenn Sie ein Szenario haben, in dem IAM Benutzer oder ein Root-Benutzer erforderlich sind AWS-Konto, aktivieren Sie die Option MFA für zusätzliche Sicherheit. Wenn Sie festlegen möchten, MFA wann API Operationen aufgerufen werden, fügen Sie MFA Bedingungen zu Ihren Richtlinien hinzu. Weitere Informationen finden Sie unter [Konfiguration des MFA -geschützten API Zugriffs](#) im IAMBenutzerhandbuch.

Weitere Informationen zu bewährten Methoden finden Sie unter [Bewährte Sicherheitsmethoden IAM im IAM](#) Benutzerhandbuch. IAM

## Verwenden der ElastiCache Konsole

Um auf die ElastiCache Amazon-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Informationen zu den ElastiCache Ressourcen in Ihrem Verzeichnis aufzulisten und einzusehen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur Anrufe an AWS CLI oder am tätigen, keine Mindestberechtigungen für die Konsole gewähren AWS API. Erlauben Sie stattdessen nur den Zugriff auf die Aktionen, die dem API Vorgang entsprechen, den sie ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die ElastiCache Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die ElastiCache ConsoleAccess oder die ReadOnly AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie im [Benutzerhandbuch unter Hinzufügen von Berechtigungen für einen IAM](#) Benutzer.

## Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen könnten, die es IAM Benutzern ermöglicht, die Inline-Richtlinien und verwalteten Richtlinien einzusehen, die mit ihrer Benutzeridentität verknüpft sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe von oder. AWS CLI AWS API

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ViewOwnUserInfo",
```

```
 "Effect": "Allow",
 "Action": [
 "iam:GetUserPolicy",
 "iam:ListGroupsWithUser",
 "iam:ListAttachedUserPolicies",
 "iam:ListUserPolicies",
 "iam:GetUser"
],
 "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
 "Sid": "NavigateInConsole",
 "Effect": "Allow",
 "Action": [
 "iam:GetGroupPolicy",
 "iam:GetPolicyVersion",
 "iam:GetPolicy",
 "iam:ListAttachedGroupPolicies",
 "iam:ListGroupPolicies",
 "iam:ListPolicyVersions",
 "iam:ListPolicies",
 "iam:ListUsers"
],
 "Resource": "*"
}
]
```

## Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit ElastiCache und auftreten können IAM.

### Themen

- [Ich bin nicht berechtigt, eine Aktion durchzuführen in ElastiCache](#)
- [Ich bin nicht berechtigt, iam auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine ElastiCache Ressourcen ermöglichen](#)

## Ich bin nicht berechtigt, eine Aktion durchzuführen in ElastiCache

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson`-Benutzer versucht, die Konsole zum Anzeigen von Details zu einer fiktiven `my-example-widget`-Ressource zu verwenden, jedoch nicht über `elasticache:GetWidget`-Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticache:GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion `my-example-widget` auf die Ressource `elasticache:GetWidget` zugreifen zu können.

## Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion auszuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an ElastiCache diese Person übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in ElastiCache auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

## Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine ElastiCache Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen ElastiCache unterstützt werden, finden Sie unter. [So ElastiCache arbeitet Amazon mit IAM](#)
- Informationen darüber, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie [im IAM Benutzerhandbuch unter Gewähren des Zugriffs auf einen anderen IAM Benutzer AWS-Konto , der Ihnen gehört.](#)
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.](#)
- Informationen dazu, wie Sie Zugriff über einen Identitätsverbund [gewähren, finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\).](#) IAM
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAM im Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff.](#) IAM

## Zugriffskontrolle

Sie können über gültige Anmeldeinformationen verfügen, um Ihre Anfragen zu authentifizieren, aber ohne die entsprechenden Berechtigungen können Sie keine ElastiCache Ressourcen erstellen oder darauf zugreifen. Sie benötigen beispielsweise die erforderlichen Berechtigungen, um einen ElastiCache Cluster zu erstellen.

In den folgenden Abschnitten wird beschrieben, wie Sie Berechtigungen für verwalten ElastiCache. Wir empfehlen Ihnen, zunächst die Übersicht zu lesen.

- [Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen](#)
- [Verwendung identitätsbasierter Richtlinien \(IAM Richtlinien\) für Amazon ElastiCache](#)

# Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen

Jede AWS Ressource gehört einem AWS Konto, und die Berechtigungen zum Erstellen oder Zugreifen auf eine Ressource werden durch Berechtigungsrichtlinien geregelt. Ein Kontoadministrator kann IAM Identitäten (d. h. Benutzern, Gruppen und Rollen) Berechtigungsrichtlinien zuordnen. Darüber hinaus unterstützt Amazon ElastiCache auch das Anhängen von Berechtigungsrichtlinien an Ressourcen.

## Note

Ein Kontoadministrator (oder Administratorbenutzer) ist ein Benutzer mit Administratorrechten. Weitere Informationen finden Sie unter [IAM Bewährte Methoden](#) im IAM Benutzerhandbuch.

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Benutzer, IAM die über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Folgen Sie den Anweisungen [unter Erstellen einer Rolle für einen externen Identitätsanbieter \(Federation\)](#) im IAM Benutzerhandbuch.

- IAM Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen [unter Eine Rolle für einen IAM Benutzer erstellen](#) im IAM Benutzerhandbuch.

- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Folgen Sie den Anweisungen [unter Hinzufügen von Berechtigungen für einen Benutzer \(Konsole\)](#) im IAM Benutzerhandbuch.

## Themen

- [ElastiCache Ressourcen und Abläufe von Amazon](#)
- [Grundlegendes zum Eigentum an Ressourcen](#)

- [Verwalten des Zugriffs auf Ressourcen](#)
- [AWS verwaltete Richtlinien für Amazon ElastiCache](#)
- [Verwendung identitätsbasierter Richtlinien \(IAMRichtlinien\) für Amazon ElastiCache](#)
- [Berechtigungen auf Ressourcenebene](#)
- [Verwenden von Bedingungsschlüssel](#)
- [Verwenden von serviceverknüpften Rollen für Amazon ElastiCache](#)
- [ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#)

## ElastiCache Ressourcen und Abläufe von Amazon

Eine Liste der ElastiCache Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

## Grundlegendes zum Eigentum an Ressourcen

Ein Ressourcenbesitzer ist das AWS Konto, das die Ressource erstellt hat. Das heißt, der Ressourcenbesitzer ist das AWS Konto der Prinzipalidentität, die die Anforderung authentifiziert, mit der die Ressource erstellt wird. Eine Prinzipalidentität kann das Stammkonto, ein IAM Benutzer oder eine IAM Rolle sein. Die Funktionsweise wird anhand der folgenden Beispiele deutlich:

- Angenommen, Sie verwenden die Root-Kontoanmeldeinformationen Ihres AWS Kontos, um einen Cache-Cluster zu erstellen. In diesem Fall ist Ihr AWS Konto der Eigentümer der Ressource. ElastiCacheIn ist die Ressource der Cache-Cluster.
- Angenommen, Sie erstellen einen IAM Benutzer in Ihrem AWS Konto und gewähren diesem Benutzer Berechtigungen zum Erstellen eines Cache-Clusters. In diesem Fall kann der Benutzer einen Cache-Cluster erstellen. Ihr AWS Konto, zu dem der Benutzer gehört, besitzt jedoch die Cache-Cluster-Ressource.
- Angenommen, Sie erstellen in Ihrem AWS Konto eine IAM Rolle mit Berechtigungen zum Erstellen eines Cache-Clusters. In diesem Fall kann jeder, der die Rolle übernehmen kann, einen Cache-Cluster erstellen. Ihr AWS Konto, zu dem die Rolle gehört, besitzt die Cache-Cluster-Ressource.

## Verwalten des Zugriffs auf Ressourcen

Eine Berechtigungsrichtlinie beschreibt, wer Zugriff auf welche Objekte hat. Im folgenden Abschnitt werden die verfügbaren Optionen zum Erstellen von Berechtigungsrichtlinien erläutert.

### Note

In diesem Abschnitt wird die Verwendung IAM im Kontext von Amazon beschrieben ElastiCache. Es enthält keine detaillierten Informationen über den IAM Service. Eine vollständige IAM Dokumentation finden Sie unter [Was ist IAM?](#) im IAM Benutzerhandbuch. Informationen zur IAM Richtliniensyntax und zu Beschreibungen finden Sie unter [AWS IAM Policy Reference](#) im IAM Benutzerhandbuch.

Mit einer IAM Identität verknüpfte Richtlinien werden als identitätsbasierte Richtlinien (IAM Richtlinien) bezeichnet. An Ressourcen angehängte Richtlinien werden als ressourcenbasierte Richtlinien bezeichnet.

### Themen

- [Identitätsbasierte Richtlinien \(Richtlinien\) IAM](#)
- [Angaben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale](#)
- [Angaben von Bedingungen in einer Richtlinie](#)

### Identitätsbasierte Richtlinien (Richtlinien) IAM

Sie können Identitäten Richtlinien zuordnen IAM. Sie können z. B. Folgendes tun:

- Anfügen einer Berechtigungsrichtlinie zu einem Benutzer oder einer Gruppe in Ihrem Konto – Ein Kontoadministrator kann eine Berechtigungsrichtlinie verwenden, die einem bestimmten Benutzer zugeordnet ist, um Berechtigungen zu erteilen. In diesem Fall verfügen die Benutzer über die erforderlichen Berechtigungen zum Erstellen einer ElastiCache Ressource, z. B. eines Cache-Clusters, einer Parametergruppe oder einer Sicherheitsgruppe.
- Einer Rolle eine Berechtigungsrichtlinie zuordnen (kontoübergreifende Berechtigungen gewähren) — Sie können einer IAM Rolle eine identitätsbasierte Berechtigungsrichtlinie zuordnen, um kontoübergreifende Berechtigungen zu gewähren. Der Administrator in Konto A kann beispielsweise wie folgt eine Rolle erstellen, um einem anderen AWS Konto (z. B. Konto B) oder einem Dienst kontoübergreifende Berechtigungen zu gewähren: AWS

1. Der Administrator von Konto A erstellt eine IAM Rolle und fügt der Rolle, die Berechtigungen für Ressourcen in Konto A gewährt, eine Berechtigungsrichtlinie hinzu.
2. Der Administrator von Konto A weist der Rolle eine Vertrauensrichtlinie zu, die Konto B als den Prinzipal identifiziert, der die Rolle übernehmen kann.
3. Der Administrator von Konto B kann dann die Berechtigungen zur Übernahme der Rolle an alle Benutzer in Konto B delegieren. Auf diese Weise können Benutzer in Konto B Ressourcen in Konto A erstellen oder darauf zugreifen. In einigen Fällen möchten Sie einem AWS Dienst möglicherweise Berechtigungen zur Übernahme der Rolle erteilen. Zum Support dieses Ansatzes kann es sich beim Prinzipal in der Vertrauensrichtlinie auch um einen AWS -Service-Prinzipal handeln.

Weitere Informationen zur Verwendung IAM zum Delegieren von Berechtigungen finden Sie unter [Zugriffsverwaltung](#) im IAMBenutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die es einem Benutzer ermöglicht, die `DescribeCacheClusters` Aktion für Ihr AWS Konto auszuführen. ElastiCache unterstützt auch die Identifizierung bestimmter Ressourcen, die die Ressource ARNs für API Aktionen verwenden. (Dieser Ansatz heißt auch Ressourcenebenen-Berechtigungen.)

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "DescribeCacheClusters",
 "Effect": "Allow",
 "Action": [
 "elasticache:DescribeCacheClusters"],
 "Resource": resource-arn
]
}
```

Weitere Informationen zur Verwendung identitätsbasierter Richtlinien mit finden Sie ElastiCache unter [Verwendung identitätsbasierter Richtlinien \(IAMRichtlinien\) für Amazon ElastiCache](#) Weitere Informationen zu Benutzern, Gruppen, Rollen und Berechtigungen finden Sie im Benutzerhandbuch unter [Identitäten \(Benutzer, Gruppen und Rollen\)](#). IAM

## Angeben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale

Für jede ElastiCache Amazon-Ressource (siehe [ElastiCache Ressourcen und Abläufe von Amazon](#)) definiert der Service eine Reihe von API Vorgängen (siehe [Aktionen](#)). ElastiCache Definiert eine Reihe von API Aktionen, die Sie in einer Richtlinie angeben können, um Berechtigungen für diese Operationen zu erteilen. Für die ElastiCache Clusterressource sind beispielsweise die folgenden Aktionen definiert: `CreateCacheCluster`, `DeleteCacheCluster`, und `DescribeCacheCluster`. Für die Ausführung eines API Vorgangs können Berechtigungen für mehr als eine Aktion erforderlich sein.

### Grundlegende Richtlinienelemente:

- **Ressource** — In einer Richtlinie verwenden Sie einen Amazon-Ressourcennamen (ARN), um die Ressource zu identifizieren, für die die Richtlinie gilt. Weitere Informationen finden Sie unter [ElastiCache Ressourcen und Abläufe von Amazon](#).
- **Aktion** – Mit Aktionsschlüsselwörtern geben Sie die Ressourcenoperationen an, die Sie zulassen oder verweigern möchten. Je nach Angabe `Effect` gewährt oder verweigert die `elasticache:CreateCacheCluster` Berechtigung dem Benutzer beispielsweise die Berechtigungen zur Durchführung des ElastiCache `CreateCacheCluster` Amazon-Vorgangs.
- **Auswirkung** – Die von Ihnen festgelegte Auswirkung, wenn der Benutzer die jeweilige Aktion anfordert – entweder „allow“ (Zugriffserlaubnis) oder „deny“ (Zugriffsverweigerung). Wenn Sie den Zugriff auf eine Ressource nicht ausdrücklich gestatten ("Allow"), wird er automatisch verweigert. Sie können den Zugriff auf eine Ressource auch explizit verweigern. So können Sie zum Beispiel sicherstellen, dass ein Benutzer nicht auf die Ressource zugreifen kann, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird.
- **Principal** — In identitätsbasierten Richtlinien (IAMRichtlinien) ist der Benutzer, dem die Richtlinie zugeordnet ist, der implizite Prinzipal. In ressourcenbasierten Richtlinien müssen Sie den Benutzer, das Konto, den Service oder die sonstige Entität angeben, die die Berechtigungen erhalten soll (gilt nur für ressourcenbasierte Richtlinien).

Weitere Informationen zur IAM Richtlinienyntax und zu deren Beschreibung finden Sie unter [AWS IAMRichtlinienreferenz im IAM Benutzerhandbuch](#).

Eine Tabelle mit allen ElastiCache API Amazon-Aktionen finden Sie unter [ElastiCache APIBerechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#).

## Angeben von Bedingungen in einer Richtlinie

Wenn Sie Genehmigungen erteilen, können Sie in der Sprache der IAM Richtlinie angeben, unter welchen Bedingungen eine Richtlinie in Kraft treten soll. Beispielsweise kann festgelegt werden, dass eine Richtlinie erst ab einem bestimmten Datum gilt. Weitere Informationen zur Angabe von Bedingungen in einer Richtlinien-sprache finden Sie unter [Bedingung](#) im IAMBenutzerhandbuch.

Bedingungen werden mithilfe vordefinierter Bedingungsschlüssel formuliert. Informationen zur Verwendung von ElastiCache -spezifischen Bedingungsschlüsseln finden Sie unter [Verwenden von Bedingungsschlüssel](#). Es gibt Bedingungsschlüssel AWS für alle Bereiche, die Sie je nach Bedarf verwenden können. Eine vollständige Liste der AWS-weiten Schlüssel finden Sie im IAMBenutzerhandbuch unter [Verfügbare Schlüssel für Bedingungen](#).

## AWS verwaltete Richtlinien für Amazon ElastiCache

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [kundenverwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [AWS Verwaltete Richtlinien](#).

### AWS verwaltete Richtlinie: ElastiCacheServiceRolePolicy

Sie können keine Verbindungen ElastiCacheServiceRolePolicy zu Ihren IAM Entitäten herstellen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es ElastiCache ermöglicht, Aktionen in Ihrem Namen durchzuführen.

Diese Richtlinie ermöglicht es ElastiCache , AWS Ressourcen in Ihrem Namen zu verwalten, sofern dies für die Verwaltung Ihres Caches erforderlich ist:

- `ec2`— Verwalten Sie EC2 Netzwerkressourcen, die an Cache-Knoten angehängt werden sollen, darunter VPC Endpunkte (für serverlose Caches), Elastic Network Interfaces (ENIs) (für selbst entworfene Cluster) und Sicherheitsgruppen.
- `cloudwatch`— Sendet metrische Daten aus dem Service in. CloudWatch

- `outposts`— Erlaubt die Erstellung von Cache-Knoten auf AWS Outposts.

Sie finden die [ElastiCacheServiceRolePolicy](#) Richtlinie auf der IAM Konsole und [ElastiCacheServiceRolePolicy](#) im Referenzhandbuch für AWS verwaltete Richtlinien.

AWS verwaltete Richtlinie: `AmazonElastiCacheFullAccess`

Sie können die `AmazonElastiCacheFullAccess` Richtlinie an Ihre IAM Identitäten anhängen.

Diese Richtlinie gewährt Prinzipalen vollen Zugriff auf die ElastiCache Nutzung der AWS Management Console:

- `elasticache`— Zugriff auf alle APIs.
- `iam` – Erstellen einer serviceverknüpften Rolle, die für den Servicebetrieb erforderlich ist.
- `ec2`— Beschreiben Sie abhängige EC2 Ressourcen VPC, die für die Cache-Erstellung erforderlich sind (Subnetz, Sicherheitsgruppe), und ermöglichen Sie die Erstellung von VPC Endpunkten (für serverlose Caches).
- `kms`— Erlaubt die Verwendung von vom Kunden verwalteten für. CMKs encryption-at-rest
- `cloudwatch`— Erlaubt den Zugriff auf Metriken, um ElastiCache Metriken in der Konsole anzuzeigen.
- `application-autoscaling` – Erlauben des Zugriffs zur Beschreibung von AutoScaling-Richtlinien für Caches.
- `logs` – Wird verwendet, um Protokollstreams für die Protokollbereitstellungsfunktion in der Konsole aufzufüllen.
- `firehose` – Wird verwendet, um Bereitstellungsdatenströme für die Protokollbereitstellungsfunktion in der Konsole aufzufüllen.
- `s3` – Wird verwendet, um S3-Buckets für die Funktion zur Wiederherstellung von Snapshots in der Konsole aufzufüllen.
- `outposts`— Wird verwendet, um AWS Outposts für die Cache-Erstellung in der Konsole zu füllen.
- `sns`— Wird verwendet, um SNS Themen für die Benachrichtigungsfunktion in der Konsole aufzufüllen.

Sie finden die [AmazonElastiCacheFullAccess](#) Richtlinie auf der IAM Konsole und [AmazonElastiCacheFullAccess](#) im Referenzhandbuch für AWS verwaltete Richtlinien.

## AWS verwaltete Richtlinie: AmazonElastiCacheReadOnlyAccess

Sie können die AmazonElastiCacheReadOnlyAccess Richtlinie an Ihre IAM Identitäten anhängen.

Diese Richtlinie gewährt Prinzipalen nur Lesezugriff auf die Management ElastiCache Console: AWS

- elasticache— Nur-Lese-Zugriff. Describe APIs

Sie finden die [AmazonElastiCacheReadOnlyAccess](#) Richtlinie auf der IAM Konsole und [AmazonElastiCacheReadOnlyAccess](#) im AWS Managed Policy Reference Guide.

### ElastiCache Aktualisierungen der AWS verwalteten Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien, die ElastiCache seit Beginn der Nachverfolgung dieser Änderungen durch diesen Dienst vorgenommen wurden. Abonnieren Sie den RSS Feed auf der Seite ElastiCache Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderung	Beschreibung	Datum
<a href="#">AmazonElastiCacheFullAccess</a> – Aktualisierung auf eine bestehende Richtlinie	ElastiCache neue Berechtigungen wurden hinzugefügt, um die Verwaltung von serverlosen Caches und die Nutzung aller Servicefunktionen über die Konsole zu ermöglichen.	8. November 2023
<a href="#">ElastiCacheServiceRolePolicy</a> – Aktualisierung auf eine bestehende Richtlinie	ElastiCache neue Berechtigungen hinzugefügt, um die Verwaltung von VPC Endpunkten für serverlose Cache-Ressourcen zu ermöglichen.	8. November 2023
ElastiCache hat begonnen, Änderungen zu verfolgen	ElastiCache hat begonnen, Änderungen für die AWS	7. Februar 2020

Änderung	Beschreibung	Datum
	verwalteten Richtlinien zu verfolgen.	

## Verwendung identitätsbasierter Richtlinien (IAMRichtlinien) für Amazon ElastiCache

Dieses Thema enthält Beispiele für identitätsbasierte Richtlinien, mit denen ein Kontoadministrator Berechtigungsrichtlinien an IAM Identitäten (d. h. Benutzern, Gruppen und Rollen) anhängen kann.

### Important

Wir empfehlen Ihnen, zunächst die Themen zu lesen, in denen die grundlegenden Konzepte und Optionen zur Verwaltung des Zugriffs auf ElastiCache Amazon-Ressourcen erläutert werden. Weitere Informationen finden Sie unter [Überblick über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache Ressourcen](#).

Dieses Thema besteht aus folgenden Abschnitten:

- [AWS verwaltete Richtlinien für Amazon ElastiCache](#)
- [Beispiele für vom Kunden verwaltete Richtlinien](#)

Im Folgenden finden Sie ein Beispiel für eine Berechtigungsrichtlinie bei der Verwendung von 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"
 }
]
}

```

Im Folgenden wird ein Beispiel für eine Berechtigungsrichtlinie bei der Verwendung von Memcached gezeigt.

```

{
 "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"
 }
]
}

```

Die Richtlinie enthält zwei Anweisungen:

- Die erste Anweisung erteilt Berechtigungen für die ElastiCache Amazon-Aktionen (`elasticache:Create*`, `elasticache:Describe*`, `elasticache:Modify*`)
- Die zweite Anweisung erteilt Berechtigungen für die IAM Aktion (`iam:PassRole`) für den IAM Rollennamen, der am Ende des Resource Werts angegeben ist.

Das Element `Principal` ist in der Richtlinie nicht angegeben, da in identitätsbasierten Richtlinien die Angabe des Prinzipals als Empfänger der Berechtigung nicht erforderlich ist. Wenn Sie einem Benutzer eine Richtlinie zuweisen, ist der Benutzer automatisch der Prinzipal. Wenn Sie einer IAM Rolle eine Berechtigungsrichtlinie zuordnen, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Eine Tabelle mit allen ElastiCache API Amazon-Aktionen und den Ressourcen, für die sie gelten, finden Sie unter [ElastiCache API Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#).

Beispiele für vom Kunden verwaltete Richtlinien

Wenn Sie keine Standardrichtlinie verwenden und eine benutzerdefinierte verwaltete Richtlinie verwenden möchten, stellen Sie eines sicher. Sie sollten entweder die Genehmigung zum Aufrufen von `iam:createServiceLinkedRole` haben (weitere Informationen finden Sie unter [Beispiel 4: Erlauben Sie einem Benutzer, anzurufen IAM CreateServiceLinkedRole API](#)). Oder Sie hätten eine ElastiCache servicebezogene Rolle erstellen sollen.

In Kombination mit den Mindestberechtigungen, die für die Nutzung der ElastiCache Amazon-Konsole erforderlich sind, gewähren die Beispielrichtlinien in diesem Abschnitt zusätzliche Berechtigungen. Die Beispiele sind auch relevant für die AWS SDKs und die AWS CLI.

Anweisungen zum Einrichten von IAM Benutzern und Gruppen finden Sie im Benutzerhandbuch unter [Erstellen Ihrer ersten IAM Benutzer- und Administratorgruppe](#). IAM

#### Important

Testen Sie Ihre IAM Richtlinien immer gründlich, bevor Sie sie in der Produktion verwenden. Einige ElastiCache Aktionen, die einfach erscheinen, können andere Aktionen erfordern, um sie zu unterstützen, wenn Sie die ElastiCache Konsole verwenden. `elasticache:CreateCacheCluster` Gewährt beispielsweise Berechtigungen zum Erstellen von ElastiCache Cache-Clustern. Um diesen Vorgang auszuführen, verwendet

die ElastiCache Konsole jedoch eine Reihe von Describe List AND-Aktionen, um Konsolenlisten aufzufüllen.

## Beispiele

- [Beispiel 1: Erlauben Sie einem Benutzer nur Lesezugriff auf Ressourcen ElastiCache](#)
- [Beispiel 2: Erlauben Sie einem Benutzer, allgemeine ElastiCache Systemadministratortasken auszuführen](#)
- [Beispiel 3: Erlauben Sie einem Benutzer den Zugriff auf alle ElastiCache API Aktionen](#)
- [Beispiel 4: Erlauben Sie einem Benutzer, anzurufen IAM CreateServiceLinkedRole API](#)
- [Beispiel 5: Erlauben Sie einem Benutzer, mithilfe von Authentifizierung eine Verbindung zum serverlosen Cache herzustellen IAM](#)

### Beispiel 1: Erlauben Sie einem Benutzer nur Lesezugriff auf Ressourcen ElastiCache

Die folgende Richtlinie gewährt ElastiCache Zugriffsaktionen, die es einem Benutzer ermöglichen, Ressourcen aufzulisten. Gewöhnlich ordnen Sie diese Art von Berechtigungsrichtlinie einer Gruppe von Managern zu.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECReadOnly",
 "Effect": "Allow",
 "Action": [
 "elasticache:Describe*",
 "elasticache:List*"
],
 "Resource": "*"
 }
]
```

### Beispiel 2: Erlauben Sie einem Benutzer, allgemeine ElastiCache Systemadministratortasken auszuführen

Zu den allgemeinen Aufgaben des Systemadministrators gehört das Ändern von Ressourcen. Ein Systemadministrator möchte möglicherweise auch Informationen über die ElastiCache Ereignisse

erhalten. Die folgende Richtlinie gewährt einem Benutzer die Berechtigung, ElastiCache Aktionen für diese allgemeinen Systemadministratortasken auszuführen. Normalerweise ordnen Sie diese Art Berechtigungsrichtlinie der Systemadministratorengruppe zu.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECAAllowMutations",
 "Effect": "Allow",
 "Action": [
 "elasticache:Modify*",
 "elasticache:Describe*",
 "elasticache:ResetCacheParameterGroup"
],
 "Resource": "*"
 }
]
```

**Beispiel 3:** Erlauben Sie einem Benutzer den Zugriff auf alle ElastiCache API Aktionen

Die folgende Richtlinie ermöglicht einem Benutzer den Zugriff auf alle ElastiCache Aktionen. Es wird empfohlen, diese Art von Berechtigungsrichtlinie nur einem Administratorbenutzer zu gewähren.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECAAllowAll",
 "Effect": "Allow",
 "Action": [
 "elasticache:*"
],
 "Resource": "*"
 }
]
```

**Beispiel 4:** Erlauben Sie einem Benutzer, anzurufen IAM CreateServiceLinkedRole API

Die folgende Richtlinie ermöglicht es dem Benutzer, die anzurufen IAM CreateServiceLinkedRoleAPI. Wir empfehlen, dass Sie dem Benutzer, der mutative ElastiCache Operationen aufruft, diese Art von Berechtigungsrichtlinie gewähren.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CreateSLRAllows",
 "Effect": "Allow",
 "Action": [
 "iam:CreateServiceLinkedRole"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "iam:AWS ServiceName": "elasticache.amazonaws.com"
 }
 }
 }
]
}
```

Beispiel 5: Erlauben Sie einem Benutzer, mithilfe von Authentifizierung eine Verbindung zum serverlosen Cache herzustellen IAM

Die folgende Richtlinie ermöglicht es jedem Benutzer, zwischen 2023-04-01 und 2023-06-30 mithilfe der IAM Authentifizierung eine Verbindung zu einem beliebigen serverlosen Cache herzustellen.

```
{
 "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:*"
]
 }
]
```

## Berechtigungen auf Ressourcenebene

Sie können den Umfang der Berechtigungen einschränken, indem Sie Ressourcen in einer IAM Richtlinie angeben. Viele ElastiCache API Aktionen unterstützen einen Ressourcentyp, der je nach Verhalten der Aktion variiert. Jede IAM Richtlinienerklärung erteilt die Genehmigung für eine Aktion, die an einer Ressource ausgeführt wird. Wenn die Aktion nicht auf eine benannte Ressource wirkt oder wenn Sie die Erlaubnis erteilen, die Aktion auf allen Ressourcen durchzuführen, ist der Wert der Ressource in der Richtlinie ein Platzhalter (\*). Bei vielen API Aktionen können Sie die Ressourcen einschränken, die ein Benutzer ändern kann, indem Sie den Amazon-Ressourcennamen (ARN) einer Ressource oder ein ARN Muster angeben, das mehreren Ressourcen entspricht. Um die Berechtigungen pro Ressource einzuschränken, geben Sie die Ressource nach anARN.

Eine Liste der ElastiCache Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

### Beispiele

- [Beispiel 1: Erlauben Sie einem Benutzer vollen Zugriff auf bestimmte ElastiCache Ressourcentypen](#)
- [Beispiel 2: Verweigern Sie einem Benutzer den Zugriff auf einen Serverless-Cache.](#)

Beispiel 1: Erlauben Sie einem Benutzer vollen Zugriff auf bestimmte ElastiCache Ressourcentypen

Die folgende Richtlinie erlaubt explizit alle Ressourcen vom Typ Serverless-Cache.

```
{
 "Sid": "Example1",
 "Effect": "Allow",
 "Action": "elasticache:*",
 "Resource": [
 "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
]
}
```

```
]
}
```

Beispiel 2: Verweigern Sie einem Benutzer den Zugriff auf einen Serverless-Cache.

Im folgenden Beispiel wird der Zugriff auf einen bestimmten Serverless-Cache explizit verweigert.

```
{
 "Sid": "Example2",
 "Effect": "Deny",
 "Action": "elasticache:*",
 "Resource": [
 "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
]
}
```

## Verwenden von Bedingungsschlüssel

Sie können Bedingungen angeben, die bestimmen, wie eine IAM Richtlinie wirksam wird. In können Sie das Condition Element einer JSON Richtlinie verwenden ElastiCache, um Schlüssel im Anforderungskontext mit Schlüsselwerten zu vergleichen, die Sie in Ihrer Richtlinie angeben. Weitere Informationen finden Sie unter [IAMJSONRichtlinienelemente: Bedingung](#).

Eine Liste der ElastiCache Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference.

Eine Liste der globalen Bedingungsschlüssel finden Sie unter [AWS Globale Bedingungskontextschlüssel](#).

### Festlegung von Bedingungen: Verwenden von Bedingungsschlüsseln

Um eine differenzierte Steuerung zu implementieren, schreiben Sie eine IAM Berechtigungsrichtlinie, die Bedingungen für die Steuerung einzelner Parameter bei bestimmten Anfragen festlegt. Anschließend wenden Sie die Richtlinie auf IAM Benutzer, Gruppen oder Rollen an, die Sie mit der IAM Konsole erstellen.

Um eine Bedingung anzuwenden, fügen Sie die Bedingungsinformationen zur IAM Richtlinienerklärung hinzu. Im folgenden Beispiel geben Sie die Bedingung an, dass jeder erstellte selbst entworfene Cache-Cluster vom Knotentyp `cache.r5.large` ist.

Im Folgenden wird ein Beispiel für diese Berechtigungsrichtlinie bei der Verwendung von Valkey oder OSS Redis gezeigt.

```

{
 "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"
]
 }
 }
 }
]
}

```

Im Folgenden wird ein Beispiel für diese Berechtigungsrichtlinie bei der Verwendung von Memcached gezeigt.

```

{
 "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"
]
 }
 }
 }
]
}

```

Weitere Informationen finden Sie unter [Verschlagworten Sie Ihre Ressourcen ElastiCache](#) .

Weitere Informationen zur Verwendung von Richtlinienbedingungsoperatoren finden Sie unter [ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#).

Beispielrichtlinien: Verwenden von Bedingungen für die differenzierte Parameterkontrolle

Dieser Abschnitt zeigt Beispielrichtlinien für die Implementierung einer detaillierten Zugriffskontrolle für die zuvor aufgeführten Parameter. ElastiCache

1. elasticache:MaximumDataStorage: Geben Sie den maximalen Datenspeicher eines serverlosen Caches an. Unter den angegebenen Bedingungen kann der Kunde keine Caches erstellen, die mehr als eine bestimmte Datenmenge speichern können.

```

{
 "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:MaximumECPUPer Second: Geben Sie den Maximalwert ECPU pro Sekunde für einen serverlosen Cache an. Unter Verwendung der bereitgestellten Bedingungen kann der Kunde keine Caches erstellen, die mehr als eine bestimmte Anzahl von Caches pro Sekunde ausführen können. 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`: Geben Sie an, welche `NodeType` (s) ein Benutzer erstellen kann. Unter Verwendung der bereitgestellten Bedingungen kann der Kunde einen einzelnen Wert oder einen Bereichswert für einen Knotentyp angeben.

```

{
 "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`: Geben Sie mit Memcached an, welche `NodeType` (s) ein Benutzer erstellen kann. Unter Verwendung der bereitgestellten Bedingungen kann der Kunde einen einzelnen Wert oder einen Bereichswert für einen Knotentyp angeben.

```

{
 "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`: Erstellt eine Replikationsgruppe mit weniger als 20 Knotengruppen.

```

{
 "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`: Geben Sie die Replikate pro Knoten zwischen 5 und 10 an.

```

{
 "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: Geben Sie die Verwendung der Engine-Version 5.0.6 an.

```
{
 "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: Gibt die Verwendung der Memcached-Engine-Version 1.6.6 an

```
{
 "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`: Geben Sie an, dass nur eine Valkey- oder Redis-Engine verwendet wird.  
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"
 }
 }
 }
]
}

```

`elasticache:AtRestEncryptionEnabled`: Geben Sie an, dass Replikationsgruppen nur bei aktivierter Verschlüsselung erstellt werden.

```

{
 "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

- Setzen Sie den `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `false` für die [CreateReplicationGroup](#) Aktion auf, um anzugeben, dass Replikationsgruppen nur erstellt werden können, wenn TLS sie nicht verwendet werden:

```

{
 "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"
 }
}
]
}

```

Wenn der `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `false` in einer Richtlinie für die [CreateReplicationGroup](#) Aktion auf gesetzt ist, TLS ist eine `CreateReplicationGroup` Anfrage nur zulässig, wenn sie nicht verwendet wird (d. h. wenn die Anforderung keinen Parameter enthält, der auf gesetzt ist, `true` oder einen `TransitEncryptionEnabled TransitEncryptionMode` Parameter, der auf gesetzt ist)required.

- b. Setzen Sie den `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `true` für die [CreateReplicationGroup](#) Aktion auf, um anzugeben, dass Replikationsgruppen nur erstellt werden können, wenn sie TLS verwendet werden:

```

{
 "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"
 }
}
]
}

```

Wenn der `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `true` in einer Richtlinie für die [CreateReplicationGroup](#) Aktion auf gesetzt ist, ist eine `CreateReplicationGroup` Anfrage nur zulässig, wenn die Anforderung einen Parameter enthält, der auf `set` ist, `true` und einen `TransitEncryptionEnabled` Parameter, der auf `required` ist.

- c. Wird `true` für die `ModifyReplicationGroup` Aktion `elasticache:TransitEncryptionEnabled` auf gesetzt, um anzugeben, dass Replikationsgruppen nur geändert werden können, wenn TLS sie verwendet werden:

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "elasticache:ModifyReplicationGroup"
],
 "Resource": [
 "arn:aws:elasticache:*:*:replicationgroup:*"
],
 "Condition": {
 "BoolIfExists": {
 "elasticache:TransitEncryptionEnabled": "true"
 }
 }
 }
]
}

```

Wenn der `elasticache:TransitEncryptionEnabled` Bedingungsschlüssel `true` in einer Richtlinie für die [ModifyReplicationGroup](#) Aktion auf gesetzt ist, ist eine `ModifyReplicationGroup` Anfrage nur zulässig, wenn die Anforderung einen Parameter enthält, der auf `required` ist. Der auf

festgelegte `TransitEncryptionEnabled` Parameter `true` kann optional ebenfalls enthalten sein, ist in diesem Fall jedoch nicht für die Aktivierung erforderlich TLS.

12 `elasticache:AutomaticFailoverEnabled`: Geben Sie an, dass Replikationsgruppen nur bei aktiviertem automatischem Failover erstellt werden.

```
{
 "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`: Geben Sie an, dass Replikationsgruppen nicht erstellt werden können, wenn Multi-AZ deaktiviert ist.

```
{
 "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`: Geben Sie an, dass Replikationsgruppen nur bei aktiviertem Clustermodus erstellt werden können.

```

{
 "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`: Geben Sie an, dass Replikationsgruppen nur mit aktiviertem AUTH Token erstellt werden können.

```

{
 "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:** Geben Sie die Anzahl der Tage (oder min/max.) an, für die der Snapshot aufbewahrt werden soll. Die folgende Richtlinie erzwingt die Speicherung von Backups für mindestens 30 Tage.

```

{
 "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`: Spezifiziert die Verwendung von vom Kunden verwalteten Schlüsseln.  
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`: Geben Sie eine nicht standardmäßige Parametergruppe mit spezifischen Parametern von einer Organisation in Ihren Clustern an. Sie können auch ein Benennungsmuster für Ihre Parametergruppen angeben oder das Löschen eines bestimmten Parametergruppennamens blockieren. Im Folgenden finden Sie ein Beispiel, das die Verwendung von nur `"my-org-param-group"` einschränkt.

```

{
 "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`: Geben Sie mit Memcached eine nicht standardmäßige Parametergruppe mit spezifischen Parametern von einer Organisation in Ihren Clustern an. Sie können auch ein Benennungsmuster für Ihre Parametergruppen angeben oder das Löschen eines bestimmten Parametergruppennamens blockieren. Im Folgenden finden Sie ein Beispiel, das die Verwendung von nur "" einschränkt. `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"
 }
 }
 }
]
}

```

## 20elasticacheCreateCacheCluster: CreateCacheCluster Aktion wird verweigert, wenn das Anforderungs-Tag Project fehlt oder nicht gleich oder ist. 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: Zulassen CreateCacheCluster mit cacheNodeType cache.r5.large oder cache.r6g.4xlarge und 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:: Zulassen mit cache.r5.large oder cache.r6g.4xlarge und Tag. CacheNodeType  
CreateCacheCluster cacheNodeType 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**

Beim Erstellen von Richtlinien zum gemeinsamen Erzwingen von Tags und anderen Bedingungsschlüsseln kann die Bedingung `IfExists` aufgrund der zusätzlichen `elasticache:AddTagsToResource`-Richtlinienanforderungen für Erstellungsanforderungen mit dem `--tags`-Parameter für Bedingungsschlüsselemente erforderlich sein.

## Verwenden von serviceverknüpften Rollen für Amazon ElastiCache

Amazon ElastiCache verwendet [serviceverknüpfte Rollen AWS Identity and Access Management \(IAM\)](#). Eine serviceverknüpfte Rolle ist ein einzigartiger IAM Rollentyp, der direkt mit einem AWS Service wie Amazon ElastiCache verknüpft ist. Rollen, die mit dem ElastiCache Service von Amazon verknüpft sind, werden von Amazon ElastiCache vordefiniert. Sie enthalten alle Berechtigungen, die der Dienst benötigt, um AWS -Dienste im Namen Ihrer Cluster aufzurufen.

Eine serviceverknüpfte Rolle ElastiCache erleichtert die Einrichtung von Amazon, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Die Rollen sind bereits in Ihrem AWS Konto vorhanden, sind jedoch mit ElastiCache Amazon-Anwendungsfällen verknüpft und haben vordefinierte Berechtigungen. Nur Amazon ElastiCache kann diese Rollen übernehmen, und nur diese Rollen können die vordefinierten Berechtigungsrichtlinien verwenden. Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dies schützt Ihre ElastiCache Amazon-Ressourcen, da Sie die für den Zugriff auf die Ressourcen erforderlichen Berechtigungen nicht versehentlich entfernen können.

Informationen zu anderen Diensten, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS Services, die mit Services funktionieren](#). Suchen Sie in der IAM Spalte Serviceverknüpfte Rolle nach den Diensten, für die Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

### Inhalt

- [Servicebezogene Rollenberechtigungen für Amazon ElastiCache](#)
  - [Berechtigungen zum Erstellen einer serviceverknüpften Rolle](#)
- [Eine serviceverknüpfte Rolle erstellen \(\) IAM](#)

- [Eine dienstverknüpfte Rolle erstellen \(Konsole\) IAM](#)
- [Eine dienstverknüpfte Rolle erstellen \(\) IAM CLI](#)
- [Eine dienstverknüpfte Rolle erstellen \(\) IAM API](#)
- [Beschreibung einer serviceverknüpften Rolle für Amazon bearbeiten ElastiCache](#)
  - [Bearbeiten einer servicebezogenen Rollenbeschreibung \(IAMKonsole\)](#)
  - [Eine Beschreibung einer dienstbezogenen Rolle bearbeiten \(\) IAM CLI](#)
  - [Beschreibung einer serviceverknüpften Rolle bearbeiten \(\) IAM API](#)
- [Löschen einer serviceverknüpften Rolle für Amazon ElastiCache](#)
  - [Bereinigen einer serviceverknüpften Rolle](#)
  - [Löschen einer serviceverknüpften Rolle \(IAMKonsole\)](#)
  - [Löschen einer dienstverknüpften Rolle \(\) IAM CLI](#)
  - [Löschen einer serviceverknüpften Rolle \(\) IAM API](#)

## Servicebezogene Rollenberechtigungen für Amazon ElastiCache

### Berechtigungen zum Erstellen einer serviceverknüpften Rolle

Um es einer IAM Entität zu ermöglichen, eine AWS ServiceRoleForElastiCache servicebezogene Rolle zu erstellen

Fügen Sie den Berechtigungen für diese IAM Entität die folgende Richtlinienerklärung hinzu:

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

Um einer IAM Entität das Löschen einer AWS ServiceRoleForElastiCache dienstbezogenen Rolle zu ermöglichen

Fügen Sie den Berechtigungen für diese IAM Entität die folgende Richtlinienerklärung hinzu:

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

Alternativ können Sie eine AWS verwaltete Richtlinie verwenden, um vollen Zugriff auf Amazon zu gewähren ElastiCache.

Eine serviceverknüpfte Rolle erstellen ( ) IAM

Sie können eine dienstbezogene Rolle mithilfe der IAM KonsoleCLI, oder erstellen. API

Eine dienstverknüpfte Rolle erstellen (Konsole) IAM

Sie können die IAM Konsole verwenden, um eine dienstbezogene Rolle zu erstellen.

So erstellen Sie eine serviceverknüpfte Rolle (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich der IAM Konsole Rollen aus. Klicken Sie auf Create New Role (Neue Rolle erstellen).
3. Wählen Sie unter Select type of trusted entity (Typ der vertrauenswürdigen Entität auswählen) die Option AWS -Service aus.
4. Wählen Sie unter Oder wählen Sie einen Dienst aus, um seine Anwendungsfälle anzuzeigen, die Option ElastiCache.
5. Wählen Sie Weiter: Berechtigungen aus.
6. Beachten Sie unter Richtliniennamen, dass ElastiCacheServiceRolePolicy für diese Rolle erforderlich ist. Wählen Sie Weiter: Tags aus.
7. Beachten Sie, dass Tags für serviceverknüpfte Rollen nicht unterstützt werden. Klicken Sie auf Next:Review (Weiter: Prüfen).
8. (Optional:) Bearbeiten Sie in Role description die Beschreibung für die neue serviceverknüpfte Rolle.

9. Prüfen Sie die Rolle und klicken Sie dann auf Create Role (Rolle erstellen).

Eine dienstverknüpfte Rolle erstellen () IAM CLI

Sie können IAM Operationen von verwenden, AWS Command Line Interface um eine serviceverknüpfte Rolle zu erstellen. Diese Rolle kann die Vertrauensrichtlinie, sowie die enthaltenen Richtlinien enthalten, die der Service für die Zuweisung der Rolle benötigt.

Um eine dienstverknüpfte Rolle zu erstellen () CLI

Führen Sie die folgenden Operationen aus:

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

Eine dienstverknüpfte Rolle erstellen () IAM API

Sie können den verwenden IAMAPI, um eine serviceverknüpfte Rolle zu erstellen. Diese Rolle kann die Vertrauensrichtlinie, sowie die enthaltenen Richtlinien enthalten, die der Service für die Zuweisung der Rolle benötigt.

Um eine dienstverknüpfte Rolle zu erstellen () API

Verwenden Sie den [CreateServiceLinkedRole](#) APIanrufen. Geben Sie in der Anforderung einen Servicenamen im Format `elasticache.amazonaws.com` an.

Beschreibung einer serviceverknüpften Rolle für Amazon bearbeiten ElastiCache

Amazon ElastiCache erlaubt Ihnen nicht, die AWS ServiceRoleForElastiCache serviceverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können die Beschreibung der Rolle jedoch mithilfe IAM von bearbeiten.

Bearbeiten einer servicebezogenen Rollenbeschreibung (IAMKonsole)

Sie können die IAM Konsole verwenden, um eine Beschreibung einer dienstbezogenen Rolle zu bearbeiten.

So bearbeiten Sie die Beschreibung einer serviceverknüpften Rolle (Konsole)

1. Wählen Sie im Navigationsbereich der IAM Konsole die Option Rollen aus.

2. Wählen Sie den Namen der zu ändernden Rolle.
3. Wählen Sie neben Role description ganz rechts Edit.
4. Geben Sie eine neue Beschreibung im Dialogfeld ein und klicken Sie auf Save (Speichern).

Eine Beschreibung einer dienstbezogenen Rolle bearbeiten () IAM CLI

Sie können IAM Operationen von verwenden AWS Command Line Interface , um eine servicebezogene Rollenbeschreibung zu bearbeiten.

Um die Beschreibung einer dienstbezogenen Rolle zu ändern () CLI

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie den IAM Vorgang AWS CLI [get-role](#) for.

Example

```
$ aws iam get-role --role-name AWS ServiceRoleForElastiCache
```

Verwenden Sie den Rollennamen, nicht denARN, um auf Rollen mit den CLI Operationen zu verweisen. Wenn eine Rolle beispielsweise Folgendes hatARN:arn:aws:iam::123456789012:role/myrole, bezeichnen Sie die Rolle als**myrole**.

2. Um die Beschreibung einer serviceverknüpften Rolle zu aktualisieren, verwenden Sie den IAM Vorgang AWS CLI [update-role-description](#) for.

Für Linux, macOS oder Unix:

```
$ aws iam update-role-description \
 --role-name AWS ServiceRoleForElastiCache \
 --description "new description"
```

Für Windows:

```
$ aws iam update-role-description ^\
 --role-name AWS ServiceRoleForElastiCache ^\
 --description "new description"
```

## Beschreibung einer serviceverknüpften Rolle bearbeiten () IAM API

Sie können die verwenden IAMAPI, um eine mit einem Dienst verknüpfte Rollenbeschreibung zu bearbeiten.

Um die Beschreibung einer dienstbezogenen Rolle zu ändern () API

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie den Vorgang IAM API [GetRole](#).

### Example

```
https://iam.amazonaws.com/
?Action=GetRole
&RoleName=AWS ServiceRoleForElastiCache
&Version=2010-05-08
&AUTHPARAMS
```

2. Verwenden Sie den IAM API Vorgang, um die Beschreibung einer Rolle zu aktualisieren [UpdateRoleDescription](#).

### Example

```
https://iam.amazonaws.com/
?Action=UpdateRoleDescription
&RoleName=AWS ServiceRoleForElastiCache
&Version=2010-05-08
&Description="New description"
```

## Löschen einer serviceverknüpften Rolle für Amazon ElastiCache

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie löschen können.

Amazon ElastiCache löscht die serviceverknüpfte Rolle nicht für Sie.

## Bereinigen einer serviceverknüpften Rolle

Bevor Sie eine serviceverknüpfte Rolle löschen können, stellen Sie zunächst sicher, dass der Rolle keine Ressourcen (Cluster oder Replikationsgruppen) zugeordnet sind. IAM

Um zu überprüfen, ob die mit dem Dienst verknüpfte Rolle über eine aktive Sitzung in der Konsole verfügt IAM

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich der IAM Konsole Rollen aus. Wählen Sie dann den Namen (nicht das Kontrollkästchen) der AWS ServiceRoleForElastiCache Rolle aus.
3. Wählen Sie auf der Seite Summary für die ausgewählte Rolle die Registerkarte Access Advisor.
4. Überprüfen Sie auf der Registerkarte Access Advisor die jüngsten Aktivitäten für die serviceverknüpfte Rolle.

Um ElastiCache Amazon-Ressourcen zu löschen, die AWS ServiceRoleForElastiCache

- Informationen zum Löschen eines Clusters finden Sie unter:
  - [Unter Verwendung der AWS Management Console](#)
  - [Verwenden Sie den AWS CLI , um einen ElastiCache Cluster zu löschen](#)
  - [Mit dem ElastiCache API](#)
- Informationen zum Löschen einer Replikationsgruppe finden Sie unter:
  - [Löschen einer Replikationsgruppe \(Konsole\)](#)
  - [Löschen einer Replikationsgruppe \(AWS CLI\)](#)
  - [Löschen einer Replikationsgruppe \(ElastiCache API\)](#)

## Löschen einer serviceverknüpften Rolle (IAMKonsole)

Sie können die IAM Konsole verwenden, um eine dienstverknüpfte Rolle zu löschen.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter <https://console.aws.amazon.com/iam/>.

2. Wählen Sie im Navigationsbereich der IAM Konsole Rollen aus. Aktivieren Sie dann das Kontrollkästchen neben dem Rollennamen, den Sie löschen möchten, nicht den Namen oder die Zeile selbst.
3. Wählen Sie für Role actions oben auf der Seite Delete role aus.
4. Überprüfen Sie im Bestätigungsdialogfeld die Daten, auf die der Dienst zuletzt zugegriffen hat. Aus diesen Daten geht hervor, wann jede der ausgewählten Rollen zuletzt auf einen AWS Dienst zugegriffen hat. Auf diese Weise können Sie leichter bestätigen, ob die Rolle derzeit aktiv ist. Wenn Sie fortfahren möchten, wählen Sie Yes, Delete aus, um die serviceverknüpfte Rolle zur Löschung zu übermitteln.
5. Sehen Sie sich die IAM Konsolenbenachrichtigungen an, um den Fortschritt beim Löschen der dienstbezogenen Rolle zu verfolgen. Da das Löschen der IAM dienstbezogenen Rolle asynchron erfolgt, kann die Löschaufgabe erfolgreich sein oder fehlschlagen, nachdem Sie die Rolle zum Löschen eingereicht haben. Wenn der Vorgang fehlschlägt, können Sie in den Benachrichtigungen View details oder View Resources auswählen, um zu erfahren, warum die Löschung fehlgeschlagen ist.

## Löschen einer dienstverknüpften Rolle () IAM CLI

Sie können IAM Operationen von verwenden, AWS Command Line Interface um eine dienstverknüpfte Rolle zu löschen.

## Um eine dienstverknüpfte Rolle zu löschen () CLI

1. Wenn Sie den Namen der serviceverknüpften Rolle, die Sie löschen möchten, nicht kennen, geben Sie den folgenden Befehl ein. Dieser Befehl listet die Rollen und ihre Amazon-Ressourcennamen (ARNs) in Ihrem Konto auf.

```
$ aws iam get-role --role-name role-name
```

Verwenden Sie den Rollennamen, nicht denARN, um auf Rollen mit den CLI Vorgängen zu verweisen. Wenn eine Rolle beispielsweise den hat ARNarn:aws:iam::123456789012:role/myrole, bezeichnen Sie die Rolle als**myrole**.

2. Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die `deletion-task-id` aus der Antwort, um den Status der Löschaufgabe zu überprüfen.

Geben Sie Folgendes ein, um eine Anforderung zum Löschen einer serviceverknüpften Rolle abzusenden.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Geben Sie Folgendes ein, um den Status der Löschaufgabe zu überprüfen.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Der Status der Löschaufgabe kann NOT\_STARTED, IN\_PROGRESS, SUCCEEDED oder FAILED lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

## Löschen einer serviceverknüpften Rolle () IAM API

Sie können die verwenden IAMAPI, um eine dienstverknüpfte Rolle zu löschen.

Um eine dienstverknüpfte Rolle zu löschen () API

1. Um einen Löschantrag für eine dienstbezogene Rolle einzureichen, rufen Sie [DeleteServiceLinkedRole](#). Geben Sie in der Anfrage einen Rollennamen an.

Da eine serviceverknüpfte Rolle nicht gelöscht werden kann, wenn sie verwendet wird oder ihr Ressourcen zugeordnet sind, müssen Sie eine Löschanforderung übermitteln. Diese Anforderung kann verweigert werden, wenn diese Bedingungen nicht erfüllt sind. Sie benötigen die DeletionTaskId aus der Antwort, um den Status der Löschaufgabe zu überprüfen.

2. Rufen Sie an, um den Status des Löschvorgangs zu überprüfen [GetServiceLinkedRoleDeletionStatus](#). Geben Sie in der Anfrage die anDeletionTaskId.

Der Status der Löschaufgabe kann NOT\_STARTED, IN\_PROGRESS, SUCCEEDED oder FAILED lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

## ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen

Wenn Sie Richtlinien für die [Zugriffskontrolle](#) einrichten und Richtlinien für Schreibberechtigungen einrichten, die an eine IAM Richtlinie angehängt werden sollen (entweder identitäts- oder ressourcenbasiert), verwenden Sie die folgende Tabelle als Referenz. In der Tabelle sind alle ElastiCache API Amazon-Operationen und die entsprechenden Aktionen aufgeführt, für die Sie Berechtigungen zur Durchführung der Aktion erteilen können. Sie geben die Aktionen im Feld `Action` der Richtlinie und einen Ressourcenwert im Feld `Resource` der Richtlinie an. Sofern nicht anders angegeben, ist die Ressource erforderlich. Einige Felder enthalten sowohl eine erforderliche Ressource als auch optionale Ressourcen. Wenn keine Ressource vorhanden ist, ist die Ressource in der Richtlinie ein Platzhalter (\*).

Sie können Bedingungsschlüssel in Ihren ElastiCache Richtlinien verwenden, um Bedingungen auszudrücken. Eine Liste mit ElastiCache spezifischen Bedingungsschlüsseln sowie den Aktionen und Ressourcentypen, für die sie gelten, finden Sie unter [Verwenden von Bedingungsschlüssel](#). Eine vollständige Liste der AWS-weiten Schlüssel finden Sie im IAM-Benutzerhandbuch unter [Kontext-Schlüssel für AWS globale Bedingungen](#).

### Note

Um eine Aktion anzugeben, verwenden Sie das `elasticache:` Präfix gefolgt vom API Operationsnamen (z. B. `elasticache:DescribeCacheClusters`).

Eine Liste der ElastiCache Aktionen finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#) in der Service Authorization Reference.

## Konformitätsvalidierung für Amazon ElastiCache

Externe Prüfer bewerten die Sicherheit und Konformität von AWS Dienstleistungen im Rahmen mehrerer AWS Compliance-Programme, wie z. B. SOC PCIRAMP, Fed und HIPAA.

Um zu erfahren, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter heruntergeladenen AWS Artifact. Weitere Informationen finden Sie unter [Berichte heruntergeladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- [Schnellstartanleitungen zu Sicherheit und Compliance](#) — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- [Architecting for HIPAA Security and Compliance on Amazon Web Services](#) — In diesem Whitepaper wird beschrieben, wie Unternehmen Anwendungen erstellen HIPAA können, die AWS für sie in Frage kommen.

 Note

Nicht alle sind berechtigt AWS-Services . HIPAA Weitere Informationen finden Sie in der [Referenz für HIPAA qualifizierte Dienste](#).

- [AWS Ressourcen zur AWS](#) von Vorschriften — Diese Sammlung von Arbeitsmapen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien für Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zusammengefasst.
- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Dies AWS-Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen zu

erfüllen PCIDSS, z. B. durch die Erfüllung der Anforderungen zur Erkennung von Eindringlingen, die in bestimmten Compliance-Frameworks vorgeschrieben sind.

- [AWS Audit Manager](#)— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

## Weitere Informationen

Allgemeine Informationen zur AWS Cloud-Compliance finden Sie im Folgenden:

- [FIPSEndpunkte nach Dienst](#)
- [Service-Updates in ElastiCache](#)
- [AWS Cloud-Konformität](#)
- [Modell der übergreifenden Verantwortlichkeit](#)
- [AWS PCIDSSCompliance-Programm](#)

## Resilienz bei Amazon ElastiCache

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Availability Zones ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Zusätzlich zur AWS globalen Infrastruktur ElastiCache bietet Amazon mehrere Funktionen, um Ihre Datenstabilität und Backup-Anforderungen zu erfüllen.

### Themen

- [Minimieren von Ausfällen](#)

## Minimieren von Ausfällen

Bei der Planung Ihrer ElastiCache Amazon-Implementierung sollten Sie so planen, dass Ausfälle nur minimale Auswirkungen auf Ihre Anwendung und Daten haben. In diesem Abschnitt werden verschiedene Ansätze vorgestellt, mit denen Sie Ihre Anwendung und Ihre Daten vor Ausfällen schützen können.

### Themen

- [Minimieren von Ausfällen mit Memcached](#)
- [Minimierung von Fehlern beim Ausführen von Valkey oder Redis OSS](#)
- [Empfehlungen](#)

### Minimieren von Ausfällen mit Memcached

Wenn Sie die Memcached-Engine verwenden, haben Sie folgende Möglichkeiten, um die Auswirkungen von Ausfällen möglichst gering zu halten. Es gibt zwei Arten von Ausfällen, die es zu berücksichtigen gilt: Knotenausfälle und Ausfälle einzelner Availability Zones.

#### Minimieren von Knotenausfällen

Serverless-Caches minimieren Knotenausfälle automatisch mit einer replizierten Multi-AZ-Architektur, sodass Knotenausfälle für Ihre Anwendung transparent sind. Sie sollten Ihre Cache-Daten auf mehrere Knoten verteilen, um die Auswirkungen von Knotenausfällen möglichst gering zu halten. Da selbst entworfene Cluster die Replikation nicht unterstützen, führt ein Knotenausfall zwangsläufig zu Datenverlusten in Ihrem Cluster.

Wenn Sie Ihren Memcached-Cluster erstellen, können Sie ihn mit 1 bis 60 Knoten oder mehr auf Anfrage erstellen. Wenn Sie Ihre Daten über mehrere Knoten partitionieren, gehen bei einem Knotenausfall weniger Daten verloren. Angenommen, Sie partitionieren Ihre Daten über 10 Knoten und ein einzelner Knoten speichert daher etwa 10 % der Cache-Daten. Bei einem Knotenausfall gehen in diesem Fall etwa 10 % Ihres Caches verloren und muss ersetzt werden, wenn ein neuer Knoten erstellt und bereitgestellt wird. Wenn dieselben Daten auf drei größere Knoten verteilt werden, gehen dagegen bei einem Knotenausfall etwa 33 % Ihrer Cache-Daten verloren.

Weitere Informationen zum Festlegen der Knotenanzahl in einem Memcached-Cluster erhalten Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

## Minimieren von Ausfällen einer Availability Zone

Serverless-Caches minimieren Ausfälle von Availability Zones automatisch mit einer replizierten Multi-AZ-Architektur, sodass AZ-Ausfälle für Ihre Anwendung transparent sind.

Verteilen Sie Ihre Knoten auf möglichst viele Availability Zones, um die Auswirkungen von Ausfällen einer Availability Zone in einem selbst entworfenen Cluster möglichst gering zu halten. Im unwahrscheinlichen Fall eines AZ-Fehlers verlieren Sie die Daten, die in dieser AZ zwischengespeichert sind, nicht die Daten, die in der anderen zwischengespeichert sind. AZs

### Warum so viele Knoten?

Wenn meine Region nur 3 Availability Zones bereitstellt, warum sollte ich dann mehr als 3 Knoten erstellen, da im Fall eines AZ-Ausfalls sowieso etwa ein Drittel meiner Daten verloren geht?

Das ist eine sehr gute Frage. Dabei gilt es zu bedenken, dass wir zwei unterschiedliche Ausfalltypen minimieren wollen, Ausfälle einzelner Knoten und Availability Zones. Wenn Daten auf mehrere Availability Zones verteilt werden und eine der Zonen ausfällt, gehen nur die Cache-Daten dieser AZ verloren, unabhängig von der Anzahl der Knoten. Wenn jedoch ein Knoten ausfällt, gehen weniger Daten verloren, je mehr Knoten Sie verwenden.

Es gibt keine "Zauberformel", anhand derer sich die beste Anzahl an Knoten pro Cluster bestimmen lässt. Wägen Sie die Auswirkungen von Datenverlusten gegen die Wahrscheinlichkeit eines Ausfalls und die Kosten ab und treffen Sie anhand dieser Faktoren eine Entscheidung.

Weitere Informationen zum Festlegen der Knotenanzahl in einem Memcached-Cluster erhalten Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Regionen und Availability Zones](#).

## Minimierung von Fehlern beim Ausführen von Valkey oder Redis OSS

Wenn Sie eine Valkey- oder OSS Redis-Engine ausführen, haben Sie die folgenden Optionen, um die Auswirkungen eines Knoten- oder Availability Zone-Ausfalls zu minimieren.

### Minimieren von Knotenausfällen

Serverless-Caches minimieren Knotenausfälle automatisch mit einer Multi-AZ-Architektur, sodass Knotenausfälle für Ihre Anwendung transparent sind. Selbst entworfene Cluster müssen entsprechend konfiguriert werden, um Ausfälle eines einzelnen Knotens zu minimieren.

Um die Auswirkungen von Valkey- oder OSS Redis-Knotenausfällen auf selbst entworfene Cluster zu minimieren, haben Sie die folgenden Optionen:

## Themen

- [Behebung von Ausfällen: Valkey- oder Redis-Replikationsgruppen OSS](#)

### Behebung von Ausfällen: Valkey- oder Redis-Replikationsgruppen OSS

Eine Valkey- oder OSS Redis-Replikationsgruppe besteht aus einem einzigen primären Knoten, von dem Ihre Anwendung sowohl lesen als auch schreiben kann, sowie aus 1 bis 5 schreibgeschützten Replikatknoten. Wenn Daten in den Primärknoten geschrieben werden, werden diese asynchron auf die Lesereplikat-Knoten aktualisiert.

Wenn ein Lesereplikat ausfällt,

1. ElastiCache erkennt das fehlgeschlagene Lesereplikat.
2. ElastiCache schaltet den ausgefallenen Knoten aus.
3. ElastiCache startet und stellt einen Ersatzknoten in derselben AZ bereit.
4. Der neue Knoten wird mit dem primären Knoten synchronisiert.

Währenddessen kann die Anwendung weiterhin Lese- und Schreibvorgänge auf den anderen Knoten ausführen.

### Valkey oder OSS Redis Multi-AZ

Sie können Multi-AZ in Ihren Valkey- oder Redis-Replikationsgruppen aktivieren. OSS Unabhängig davon, ob Sie Multi-AZ aktivieren oder nicht, werden Ausfälle des primären Knotens erkannt und dieser wird automatisch ersetzt. Der genaue Vorgang ist dabei abhängig davon, ob Multi-AZ aktiviert ist.

Wenn Multi-AZ aktiviert ist

1. ElastiCache erkennt den Ausfall des Primärknotens.
2. ElastiCache stuft den Read-Replica-Knoten mit der geringsten Replikationsverzögerung zum Primärknoten hoch.
3. Die anderen Replikate synchronisieren sich mit dem neuen primären Knoten.
4. ElastiCache erstellt eine Read Replica in der AZ des ausgefallenen Primärservers.

5. Der neue Knoten synchronisiert sich mit dem neu ernannten primären Knoten.

Das Failover zu einem Replikationsknoten erfolgt in der Regel schneller als das Erstellen und Bereitstellen eines neuen primären Knotens. Dadurch kann Ihre Anwendung schneller wieder auf den primären Knoten schreiben, als wenn Multi-AZ nicht aktiviert ist.

Weitere Informationen finden Sie unter [Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS](#).

Wenn Multi-AZ deaktiviert ist

1. ElastiCache erkennt einen primären Fehler.
2. ElastiCache schaltet den Primärserver offline.
3. ElastiCache erstellt einen neuen Primärknoten und stellt ihn bereit, um den ausgefallenen Primärknoten zu ersetzen.
4. ElastiCache synchronisiert den neuen Primärserver mit einem der vorhandenen Replikate.
5. Nach der Synchronisierung dient der neue Knoten als primärer Knoten des Clusters.

Während der Ausführung der Schritte 1 bis 4 dieses Vorgangs können keine Daten in den primären Knoten geschrieben werden. Die Anwendung kann jedoch weiterhin Lesezugriffe auf den Replikationsknoten ausführen.

Für zusätzlichen Schutz empfehlen wir, die Knoten in Ihrer Replikationsgruppe in verschiedenen Availability Zones ( ) AZs zu starten. Dadurch wirken sich Ausfälle einer AZ nur auf die Knoten in dieser AZ aus.

Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Minimieren von Ausfällen einer Availability Zone

Serverless-Caches minimieren Ausfälle von Availability Zones automatisch mit einer replizierten Multi-AZ-Architektur, sodass AZ-Ausfälle für Ihre Anwendung transparent sind.

Verteilen Sie Ihre Knoten für jeden Shard auf möglichst viele Availability Zones, um die Auswirkungen von Ausfällen einer Availability Zone in einem selbst entworfenen Cluster möglichst gering zu halten

Unabhängig von der Anzahl der Knoten in einem Shard führt ein katastrophaler Ausfall einer Availability Zone zu einem vollständigen Datenverlust Ihres Shards, wenn Sie Ihre Daten in nur einer

Availability Zone speichern. Wenn Sie Ihre Knoten jedoch in mehreren Zonen lokalisierenAZs, führt ein Ausfall einer beliebigen AZ dazu, dass Sie nur die Knoten in dieser AZ verlieren.

Bei einem Knotenausfall kann es zu einem Leistungsabfall kommen, da sich die Lesevorgänge nun auf weniger Knoten verteilen. Dieser Leistungsabfall bleibt bestehen, bis der ausgefallene Knoten ersetzt wurde.

Informationen zur Angabe der Availability Zones für Valkey- oder OSS Redis-Knoten finden Sie unter [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Verfügbarkeitszonen für ElastiCache](#).

## Empfehlungen

Wir empfehlen, Serverless-Caches über selbst entworfene Cluster zu erstellen, da Sie ohne zusätzliche Konfiguration automatisch eine bessere Fehlertoleranz erzielen. Beim Erstellen eines selbst entworfenen Clusters gibt es jedoch zwei Arten von Ausfällen, die es zu berücksichtigen gilt: Ausfälle einzelner Knoten und umfassendere Ausfälle einer Availability Zone. Um Datenverluste durch Ausfälle möglichst gering zu halten, sollten Sie beiden Arten von Ausfällen vorbeugen.

### Minimieren der Auswirkungen von Knotenausfällen

Um die Auswirkungen eines Knotenausfalls bei der Verwendung von Valkey oder Redis zu minimieren, empfehlen wirOSS, dass Ihre Implementierung mehrere Knoten in jedem Shard verwendet und die Knoten auf mehrere Availability Zones verteilt. Dies erfolgt automatisch für Serverless-Caches.

Für selbst entworfene Cluster auf Valkey oder Redis empfehlen wirOSS, Multi-AZ in Ihrer Replikationsgruppe zu aktivieren, sodass bei einem Ausfall des primären Knotens automatisch ein Failover auf ein Replikat durchgeführt ElastiCache wird.

Wenn Sie Memcached ausführen und die Daten über mehrere Knoten partitionieren, sinkt die Größe des Datenverlustes beim Ausfall eines Knotens mit steigender Anzahl verwendeter Knoten.

### Minimieren der Auswirkungen von Ausfällen einer Availability Zone

Knoten sollten auf möglichst viele Availability Zones verteilt werden, um die Auswirkungen von Ausfällen einer Availability Zone gering zu halten. Durch die gleichmäßige Verteilung Ihrer Knoten AZs werden die Auswirkungen im unwahrscheinlichen Fall eines AZ-Ausfalls minimiert. Dies erfolgt automatisch für Serverless-Caches.

## Weitere Vorsichtsmaßnahmen

Wenn Sie Valkey oder Redis verwenden, empfehlen wir Ihnen zusätzlich zu den oben genannten, regelmäßige Backups Ihres Clusters zu planen. Bei Backups (Snapshots) wird eine RDB-Datei erstellt, die Sie im Fall eines Ausfalls oder von Datenbeschädigungen zur Wiederherstellung Ihres Caches verwenden können. Weitere Informationen finden Sie unter [Snapshot und Wiederherstellung](#).

## Infrastruktursicherheit in AWS ElastiCache

Als verwalteter Dienst AWS ElastiCache wird er durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im [AWS Architecture Center](#) im Abschnitt Sicherheit und Compliance beschrieben sind.

Für den Zugriff ElastiCache über das Netzwerk verwenden Sie AWS veröffentlichte API Aufrufe. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Wir empfehlen TLS 1.3 oder höher. Kunden müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral Diffie-Hellman (E) oder Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) unterstützen. Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Darüber hinaus müssen Anfragen mithilfe einer Zugriffsschlüssel-ID und eines geheimen Zugriffsschlüssels, der einem Prinzipal zugeordnet ist, signiert werden. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

## Service-Updates in ElastiCache

ElastiCache überwacht automatisch Ihre Flotte von Caches, Clustern und Knoten, um Service-Updates anzuwenden, sobald sie verfügbar sind. Service-Updates für Serverless-Caches werden automatisch und transparent angewendet. Für selbst entworfene Cluster richten Sie ein vordefiniertes Wartungsfenster ein, damit diese Updates angewendet werden können. In einigen Fällen könnte dieser Ansatz jedoch zu starr sein und möglicherweise Ihre Geschäftsabläufe einschränken.

Mit Service-Updates steuern Sie, wann und welche Updates auf Ihre selbst entworfenen Cluster angewendet werden. Sie können den Fortschritt dieser Updates für Ihren ausgewählten ElastiCache Cluster auch in Echtzeit überwachen.

### Themen

- [Verwaltung von Service-Updates für selbst entworfene Cluster](#)

## Verwaltung von Service-Updates für selbst entworfene Cluster

ElastiCache Service-Updates für selbst entworfene Cluster werden regelmäßig veröffentlicht. Wenn Sie über einen oder mehrere qualifizierte, selbst entworfene Cluster für diese Service-Updates verfügen, erhalten Sie Benachrichtigungen per E-Mail/SNS, über das Personal Health Dashboard (PHD) und über CloudWatch Amazon-Events, wenn die Updates veröffentlicht werden. Die Updates werden auch auf der Seite Service Updates in der ElastiCache Konsole angezeigt. Mithilfe dieses Dashboards können Sie alle Service-Updates und deren Status für Ihre ElastiCache Flotte einsehen. Service-Updates für Serverless-Caches werden transparent angewendet und können nicht über Service-Updates verwaltet werden.

Sie legen fest, wann ein Update angewendet wird, bevor ein automatisches Update gestartet wird. Wir empfehlen dringend, dass Sie alle Updates des Typs Sicherheitsupdate so schnell wie möglich installieren, um sicherzustellen, dass Ihre ElastiCache Cluster immer über aktuelle up-to-date Sicherheitspatches verfügen.

In den folgenden Abschnitten werden diese Optionen ausführlich erörtert.

### Anwenden der Service-Updates

Sie können die Service-Updates auf Ihre Flotte anwenden, sobald diese Updates den Status available (Verfügbar) haben. Service-Updates sind kumulativ. Das bedeutet, dass bisher nicht angewendete Updates im neuesten Update enthalten sind.

Wenn für ein Service-Update die automatische Aktualisierung aktiviert ist, können Sie festlegen, dass keine Maßnahmen ergriffen werden, sobald es verfügbar ist. ElastiCache plant, das Update in einem der nächsten Wartungsfenster Ihrer Cluster nach dem Startdatum der automatischen Aktualisierung zu installieren. Sie erhalten entsprechende Benachrichtigungen für jede Stufe des Updates.

#### Note

Sie können nur die Service-Updates anwenden, die über den Status verfügbar oder geplant verfügen.

Weitere Informationen zur Überprüfung und Installation von dienstspezifischen Updates auf die entsprechenden ElastiCache Cluster finden Sie unter [Anwenden der Service-Updates mithilfe der Konsole](#)

Wenn ein neues Service-Update für einen oder mehrere Ihrer ElastiCache Cluster verfügbar ist, können Sie die ElastiCache Konsole oder verwendenAPI, AWS CLI um das Update anzuwenden. Die folgenden Abschnitte beschreiben die Optionen, die zum Anwenden von Updates genutzt werden können.

## Anwenden der Service-Updates mithilfe der Konsole

Um die Liste der Service-Updates und weitere Informationen anzuzeigen, gehen Sie zur Seite Service-Updates in der Konsole.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf Service-Updates.
3. Unter Service updates (Service-Updates) können Sie Folgendes anzeigen:
  - Name des Service-Updates: Der eindeutige Name des Service-Updates
  - Aktualisierungstyp: Typ des Service-Updates, also einer der folgenden Werte: security-update oder engine-update
  - Update Severity (Update-Dringlichkeit): gibt die Priorität an, die für die Anwendung des Updates gilt:
    - kritisch: Wir empfehlen Ihnen, dieses Update sofort (innerhalb von 14 Tagen oder weniger) anzuwenden.
    - Wichtig: Wir empfehlen Ihnen, dieses Update zu installieren, sobald Ihr Geschäftsfluss dies zulässt (innerhalb von 30 Tagen oder weniger).
    - medium: Wir empfehlen Ihnen, dieses Update so schnell wie möglich (innerhalb von 60 Tagen oder weniger) anzuwenden.
    - niedrig: Wir empfehlen Ihnen, dieses Update so schnell wie möglich (innerhalb von 90 Tagen oder weniger) anzuwenden.
  - Engine-Version: Wenn der Aktualisierungstyp Engine-Update ist, die Engine-Version, die aktualisiert wird.
  - Veröffentlichungsdatum: Datum der Veröffentlichung des Updates, ab dem es zur Anwendung auf die Cluster verfügbar ist.
  - Empfohlenes Mindesthaltbarkeitsdatum: ElastiCache Voraussichtliches Datum, bis zu dem die Updates angewendet werden sollen.
  - Status: Status des Updates, einer der folgenden Werte:

- Verfügbar: Das Update ist für die relevanten Cluster verfügbar.
  - Abgeschlossen: Das Update wurde erfolgreich angewendet.
  - cancelled (Storniert): Das Update wurde storniert und ist nicht mehr erforderlich.
  - expired (Abgelaufen): Das Update kann nicht mehr angewendet werden.
4. Wählen Sie ein einzelnes Update (nicht die Schaltfläche links davon), um Details zum Service-Update anzuzeigen.

Im Abschnitt Cluster update status (Cluster-Aktualisierungsstatus) können Sie eine Liste von Clustern anzeigen, auf denen das Service-Update nicht oder erst kürzlich angewendet wurde. Für jeden Cluster können Sie Folgendes anzeigen:

- Cluster-Name – Der Name des Clusters
- Nodes Updated (Aktualisierte Knoten): Verhältnis der individuellen Knoten in einem bestimmten Cluster, die aktualisiert wurden bzw. weiterhin für das betreffende Service-Update verfügbar sind.
- Aktualisierungs-Typ: Typ des Service-Updates, also einer der folgenden Werte: security-update oder engine-update
- Status: Status des Service-Updates auf dem Cluster, also einer der folgenden Werte:
  - verfügbar: Das Update ist für die relevanten Cluster verfügbar.
  - in Bearbeitung: Das Update wird gerade auf diesen Cluster angewendet.
  - scheduled (geplant): Das Aktualisierungsdatum wurde geplant.
  - complete (abgeschlossen): Das Update wurde erfolgreich angewendet. Cluster mit dem Status „Abgeschlossen“ werden nach Abschluss 7 Tage lang angezeigt.

Wenn Sie einen oder alle Cluster mit dem Status verfügbar oder geplant auswählen und dann auf Apply now (Jetzt anwenden) klicken, wird das Update auf diesen Clustern angewendet.

## Anwenden der Service-Updates mit dem AWS CLI

Nachdem Sie benachrichtigt wurden, dass Service-Updates verfügbar sind, können Sie sie mit der AWS CLI inspizieren und anwenden:

- Führen Sie den folgenden Befehl aus, um eine Beschreibung der verfügbaren Serviceaktualisierungen abzurufen:

```
aws elasticache describe-service-updates --service-update-status
available
```

Weitere Informationen finden Sie unter [describe-service-updates](#).

- Führen Sie den folgenden Befehl aus, um ein Service-Update auf eine Liste von Clustern anzuwenden:

```
aws elasticache batch-apply-update-action --service-update
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
cluster2
```

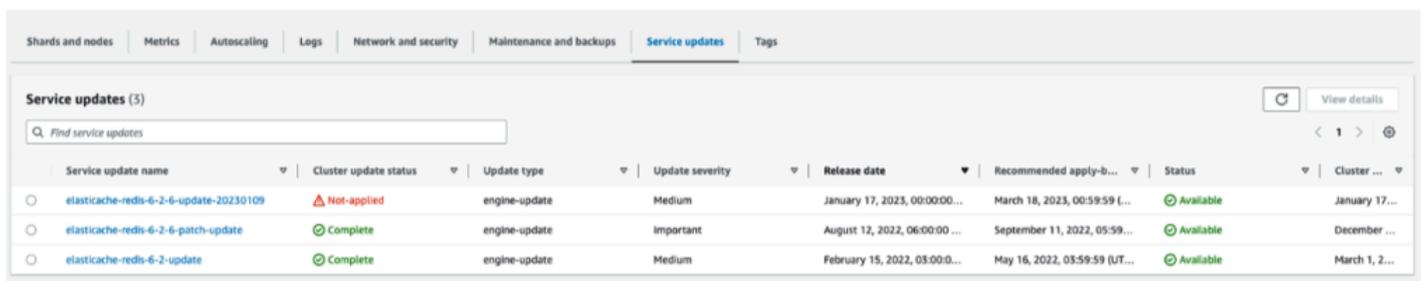
Weitere Informationen finden Sie unter [batch-apply-update-action](#).

Mithilfe der Konsole wird überprüft, ob Sie das neueste Service Update installiert haben AWS

Sie können überprüfen, ob auf Ihren ElastiCache (Redis-OSS) Clustern das neueste Service-Update ausgeführt wird, indem Sie die folgenden Schritte ausführen:

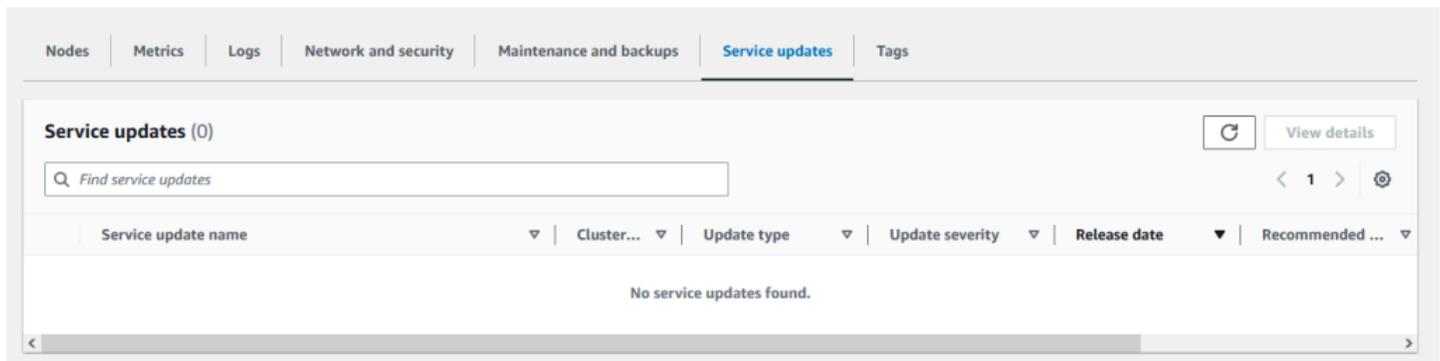
1. Wählen Sie auf der Seite OSSRedis-Cluster einen geeigneten Cluster aus
2. Wählen Sie im Navigationsbereich Service Updates aus, um die entsprechenden Service-Updates für diesen Cluster, falls vorhanden, zu sehen.

Wenn in der Konsole eine Liste mit Service-Updates angezeigt wird, können Sie das Service-Update auswählen und auf Jetzt anwenden klicken.



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, 05:59:59 (UT...	Available	March 1, 2...

Wenn auf der Konsole „Keine Dienstupdates gefunden“ angezeigt wird, bedeutet dies, dass auf dem ElastiCache (Redis-OSS) Cluster bereits das neueste Service-Update installiert wurde.



## Stoppen der Service-Updates

Sie können Updates für Cluster bei Bedarf beenden. Sie können Updates beispielsweise bei einem unerwarteten Anforderungsanstieg bei den Clustern, die aktualisiert werden, beenden. Oder Sie können Aktualisierungen beenden, wenn sie zu lange dauern und Ihren Geschäftsfluss zu Spitzenzeiten unterbrechen.

Die Operation [Stopping](#) unterbricht sofort alle Updates für die noch zu aktualisierenden Cluster und Knoten. Die Aktualisierung von Knoten mit dem Status in progress (Wird ausgeführt) wird jedoch abgeschlossen. Die Updates für andere Knoten im betreffenden Cluster, die den Status update available (Update verfügbar) aufweisen, werden dagegen nicht angewendet. Die betreffenden Knoten erhalten den Status Stopping (Wird gestoppt).

Sobald der Stopping (Wird gestoppt)-Workflow abgeschlossen ist, erhalten die Knoten mit dem Status Stopping (Wird gestoppt) den Status Stopped (Gestoppt). In Abhängigkeit vom Workflow des Updates werden in einigen Clustern keine Knoten aktualisiert. Andere Cluster enthalten dagegen bereits aktualisierte Knoten sowie Knoten, die weiterhin den Status update available (Update verfügbar) aufweisen.

Sie können den Update-Prozess später und unter Berücksichtigung der betrieblichen Anforderungen abschließen. Wählen Sie in diesem Fall die Cluster, deren Aktualisierung abgeschlossen werden soll, und dann Apply Now (Jetzt anwenden) aus. Weitere Informationen finden Sie unter [Anwenden der Service-Updates](#).

## Verwenden der Konsole

Sie können ein Service-Update über die ElastiCache Konsole unterbrechen. Das folgende Beispiel veranschaulicht dies:

- Nachdem ein Service-Update auf einem ausgewählten Cluster abgeschlossen wurde, zeigt die ElastiCache Konsole oben im Dashboard die Registerkarte „Update anzeigen/beenden“ an. ElastiCache
- Wählen Sie zum Unterbrechen des Updates Stop Update (Update stoppen) aus.
- Wählen Sie nach dem Anhalten des Updates den Cluster aus und ermitteln Sie dessen Status. Er wird mit dem Status Stopping (Wird gestoppt) und später mit dem Status Stopped (Gestoppt) angezeigt.

Unter Verwendung des AWS CLI

Sie können ein Service-Update mit der AWS CLI unterbrechen. Das folgende Codebeispiel veranschaulicht, wie dazu vorgegangen wird.

Gehen Sie für eine Replikationsgruppe wie folgt vor:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

Gehen Sie für einen Cache-Cluster wie folgt vor:

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

Weitere Informationen finden Sie unter [BatchStopUpdateAction](#).

## Häufige Sicherheitslücken und Risiken (CVE): Sicherheitslücken, die behoben wurden in ElastiCache

Common Vulnerabilities and Exposures (CVE) ist eine Liste von Einträgen für öffentlich bekannte Cybersicherheitslücken. Jeder Eintrag ist ein Link, der eine Identifikationsnummer, eine Beschreibung und mindestens eine öffentliche Referenz enthält. Auf dieser Seite finden Sie eine Liste von Sicherheitslücken, die unter behoben wurden ElastiCache.

Wir empfehlen Ihnen, immer auf die neuesten ElastiCache Valkey-, Redis OSS - oder ElastiCache Memcached-Versionen zu aktualisieren, um vor bekannten Sicherheitslücken geschützt zu sein. Wenn Sie einen ElastiCache serverlosen Cache betreiben, werden CVE Fixes automatisch auf Ihren Cache angewendet. Wenn Sie selbst entworfene Cluster mit Valkey oder Redis betreibenOSS, ElastiCache wird die Komponente verfügbar gemacht. PATCH Wenn Sie beispielsweise ElastiCache

(RedisOSS) Version 6.2.6 verwenden, ist die Hauptversion 6, die Nebenversion ist 2 und die Patch-Version ist 6. PATCHVersionen sind für abwärtskompatible Bugfixes, Sicherheitskorrekturen und nicht funktionale Änderungen vorgesehen.

Anhand der folgenden Tabelle können Sie überprüfen, ob eine bestimmte Version von ElastiCache Valkey und Redis eine Lösung für eine bestimmte OSS Sicherheitslücke enthält. Wenn auf Ihrem ElastiCache Valkey- oder OSS Redis-Cluster eine Version ohne den Sicherheitsfix ausgeführt wird, lesen Sie in der folgenden Tabelle nach und ergreifen Sie Maßnahmen. Sie können entweder ein Upgrade auf eine neuere ElastiCache Valkey- oder OSS Redis-Version durchführen, die den Fix enthält, oder wenn Sie eine Version verwenden, die den Fix enthält, stellen Sie anhand von sicher, dass Sie das neueste Service-Update installiert haben, indem Sie auf [Verwaltung von Service-Updates für selbst entworfene Cluster](#) Weitere Informationen zu den unterstützten ElastiCache Engine-Versionen und zur Durchführung eines Upgrades finden Sie unter [Motorversionen und Aufrüstung in ElastiCache](#)

#### Note

- Wenn a in einer ElastiCache Version adressiert CVE wird, bedeutet dies, dass es auch in den neueren Versionen adressiert ist. Wenn also beispielsweise eine Sicherheitslücke in ElastiCache (RedisOSS) Version 6.0.5 behoben wird, gilt dies auch für die Versionen 6.2.6, 7.0.7 und 7.1.
- Ein Sternchen (\*) in der folgenden Tabelle gibt an, dass Sie das neueste Service-Update für den (Redis-OSS) Cluster installiert haben müssen, auf dem die angegebene ElastiCache (Redis-OSS) Version ausgeführt wird, um die ElastiCache Sicherheitslücke zu beheben. Weitere Informationen darüber, wie Sie überprüfen können, ob Sie das neueste Service-Update für die ElastiCache (Redis-OSS) Version installiert haben, auf der Ihr Cluster ausgeführt wird, finden Sie unter [Verwaltung von Service-Updates für selbst entworfene Cluster](#)

ElastiCache (Redis OSS -) Version	CVEsAdressiert
Redis 6.0.5 OSS	<a href="#">CVE-2022-24735</a> *, <a href="#">-2022-24736</a> * <a href="#">CVE</a>
OSSRedis 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>

ElastiCache (Redis OSS -) Version	CVEsAdressiert
Rot OSS ist 7.0.7	<u><a href="#">CVE-2023-41056</a></u> *, <u><a href="#">CVE-2022-24834</a></u> *, <u><a href="#">-2022-35977</a></u> , <u><a href="#">-2022-36021</a></u> , <u><a href="#">-2022-24735</a></u> , <u><a href="#">-2022-24736</a></u> CVE CVE CVE CVE
OSSRot ist 7.1.0	<u><a href="#">CVE-2023-41056</a></u> , <u><a href="#">-2022-24834</a></u> , <u><a href="#">-2022-35977</a></u> , <u><a href="#">-2022-36021</a></u> , <u><a href="#">CVE-2022-24735</a></u> , <u><a href="#">-2022-24736</a></u> <u><a href="#">CVE CVE CVE CVE</a></u>

# Protokollierung und Überwachung in Amazon ElastiCache

Damit Sie Ihren Cache verwalten können, müssen Sie wissen, wie Ihre Caches funktionieren. ElastiCache generiert Metriken, die in Amazon CloudWatch Logs veröffentlicht werden, um Ihre Cache-Leistung zu überwachen. ElastiCache generiert außerdem Ereignisse, wenn signifikante Änderungen an Ihren Cache-Ressourcen vorgenommen werden (z. B. wenn ein neuer Cache erstellt oder ein Cache gelöscht wird).

## Themen

- [Serverlose Metriken und Ereignisse für Valkey und Redis OSS](#)
- [Serverlose Metriken und Ereignisse für Memcached](#)
- [ElastiCache APIAmazon-Anrufe protokollieren mit AWS CloudTrail](#)
- [SNSÜberwachung von ElastiCache Ereignissen durch Amazon](#)
- [Protokollzustellung](#)
- [Überwachung der Nutzung mit CloudWatch Metrics](#)
  
- [ElastiCache APIAmazon-Anrufe protokollieren mit AWS CloudTrail](#)

## Serverlose Metriken und Ereignisse für Valkey und Redis OSS

ElastiCache bietet eine Vielzahl von Metriken und Ereignissen zur Überwachung bei der Arbeit mit serverlosen Caches. Dazu gehören CloudWatch Metriken, Metriken auf Befehlsebene und Ereignisprotokolle, die über Amazon EventBridge aufgenommen werden können.

## Themen

- [Metriken für Serverless-Caches](#)
- [Serverless-Cache-Ereignisse](#)
- [Selbst entworfene Cluster-Metriken und Ereignisse für Valkey und Redis OSS](#)

## Metriken für Serverless-Caches

Der AWS/ElastiCache Namespace umfasst die folgenden CloudWatch Metriken für Ihre serverlosen Valkey- oder OSS Redis-Caches.

## Metrikcodes für Valkey oder Redis OSS

Metrik	Beschreibung	Einheit
BytesUsedForCache	Die Gesamtzahl der von den in Ihrem Cache gespeicherten Daten verwendeten Bytes.	Bytes
ElastiCacheProcessingUnits	Die Gesamtzahl der ElastiCacheProcessingUnits (ECPUs), die von den in Ihrem Cache ausgeführten Anfragen verbraucht wurden	Anzahl
SuccessfulReadRequestLatency	Latenz erfolgreicher Leseanfragen.	Mikrosekunden
SuccessfulWriteRequestLatency	Latenz erfolgreicher Schreibanforderungen	Mikrosekunden
TotalCmdsCount	Gesamtzahl aller Befehle, die in Ihrem Cache ausgeführt wurden	Anzahl
CacheHitRate	Gibt die Trefferrate Ihres Caches an. Dies wird mit <code>cache_hits</code> - und <code>cache_misses</code> -Statistiken auf folgende Weise berechnet : $\text{cache\_hits} / (\text{cache\_hits} + \text{cache\_misses})$ .	Prozent
CacheHits	Die Anzahl der erfolgreichen schreibgeschützten Schlüssel suchereignisse im Cache.	Anzahl
CurrConnections	Die Anzahl der Client-Verbindungen mit Ihrem Cache.	Anzahl

Metrik	Beschreibung	Einheit
ThrottledCmds	Die Anzahl der Anfragen, um die gedrosselt wurden, ElastiCache weil die Arbeitslast schneller skaliert wurde, als sie skaliert ElastiCache werden kann.	Anzahl
NewConnections	Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind.	Anzahl
Currltems	Anzahl der Elemente im Cache.	Anzahl
CurrVolatileItems	Die Anzahl der Elemente im Cache mit. TTL	Anzahl
NetworkBytesIn	Gesamtzahl der in den Cache übertragenen Bytes	Bytes
NetworkBytesOut	Gesamtzahl der aus dem Cache übertragenen Bytes	Bytes
Evictions	Die Anzahl der Schlüssel, die vom Cache bereinigt wurden	Anzahl
IamAuthenticationExpirations	Die Gesamtzahl der abgelaufenen IAM -authentifizierten Valkey- oder Redis-Verbindungen. OSS Weitere Informationen über <a href="#">Authentifizierung mit IAM</a> finden Sie im Benutzerhandbuch.	Anzahl

Metrik	Beschreibung	Einheit
IamAuthenticationThrottling	Die Gesamtzahl der gedrosselten, authentifizierten Valkey- oder Redis IAM - oder Anfragen. OSS AUTH HELLO Weitere Informationen über <a href="#">Authentifizierung mit IAM</a> finden Sie im Benutzerhandbuch.	Anzahl
KeyAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, auf Schlüssel zuzugreifen, für die sie keine Zugriffsberechtigung haben. Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl
AuthenticationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche, sich mithilfe des Befehls bei Valkey oder Redis zu authentifizieren. OSS AUTH Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl

Metrik	Beschreibung	Einheit
CommandAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, Befehle auszuführen, für deren Aufruf sie keine Berechtigung haben. Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl

### Metriken auf Befehlsebene

ElastiCache gibt auch die folgenden Metriken auf Befehlsebene aus. Gibt für jeden Befehlstyp ElastiCache die Gesamtzahl der Befehle und die Anzahl der von diesem Befehlstyp ECPUs verwendeten Befehle aus.

Metrik	Beschreibung	Einheit
EvalBasedCmds	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
EvalBasedCmdsECPUs	ECPUs von Eval-basierten Befehlen verbraucht.	Anzahl
GeoSpatialBasedCmds	Die Gesamtzahl der Befehle für raumbezogene Befehle. Dies wird aus der Commandstats-Statistik von Valkey oder OSS Redis abgeleitet. Die Ableitung erfolgt durch Summieren aller Befehle des Geo-Typs: geoad, geodist, geohash, geopos, georadius und georadiusbymember.	Anzahl

Metrik	Beschreibung	Einheit
GeoSpatialBasedCmdsECPUs	ECPUs wird von Befehlen verwendet, die auf Geodaten basieren.	Anzahl
GetTypeCmds	Die Gesamtzahl der schreibgeschützten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle mit Schreibschutz (get, hget, scard, lrange usw.) summiert werden.	Anzahl
GetTypeCmdsECPUs	ECPUs wird von Lesebefehlen verbraucht.	Anzahl
HashBasedCmds	Gesamtanzahl der Hash-basierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Hashes einwirken (hget, hkeys, hvals, hdel usw.).	Anzahl
HashBasedCmdsECPUs	ECPUs wird von Hash-basierten Befehlen verbraucht.	Anzahl

Metrik	Beschreibung	Einheit
HyperLogLogBasedCmds	Die Gesamtzahl der HyperLogLog basierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle vom Typ pf (pfadd, pfcoun, pfmerge usw.) summiert werden.	Anzahl
HyperLogLogBasedCmdsECPUs	ECPUs HyperLogLog wird von basierten Befehlen verbraucht.	Anzahl
JsonBasedCmds	Die Gesamtzahl der JSON Befehle, einschließlich Lese- und Schreibbefehlen. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle JSON Befehle summiert werden, die auf Tasten wirken. JSON	Anzahl
JsonBasedCmdsECPUs	ECPUs wird von allen JSON Befehlen, einschließlich Lese- und Schreibbefehlen, verbraucht.	Anzahl

Metrik	Beschreibung	Einheit
JsonBasedGetCmds	Die Gesamtzahl der JSON schreibgeschützten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle JSON Lesebefehle summiert werden, die sich auf Schlüssel auswirken. JSON	Anzahl
JsonBasedGetCmdsECPUs	ECPUs wird von schreibgeschützten Befehlen verbraucht. JSON	Anzahl
JsonBasedSetCmds	Die Gesamtzahl der JSON Schreibbefehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle JSON Schreibbefehle summiert werden, die auf Schlüssel einwirken. JSON	Anzahl
JsonBasedSetCmdsECPUs	ECPUs wird von Schreibbefehlen verbraucht. JSON	Anzahl
KeyBasedCmds	Gesamtanzahl der schlüsselbasierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die sich auf einen oder mehrere Schlüssel in mehreren Datenstrukturen auswirken (del, expire, rename usw.).	Anzahl

Metrik	Beschreibung	Einheit
KeyBasedCmdsECPUs	ECPUs wird von schlüsselbasierten Befehlen verbraucht.	Anzahl
ListBasedCmds	Gesamtanzahl der listenbasierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf eine oder mehrere Listen (lindex, lrange, lpush, ltrim usw.) einwirken.	Anzahl
ListBasedCmdsECPUs	ECPUs wird von listenbasierten Befehlen verbraucht.	Anzahl
NonKeyTypeCmds	Gesamtanzahl der nicht schlüsselbasierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die nicht auf einen Schlüssel einwirken, z. B. acl, dbsize oder info.	Anzahl
NonKeyTypeCmdsECPUs	ECPUs non-key-based von Befehlen verbraucht.	Anzahl

Metrik	Beschreibung	Einheit
PubSubBasedCmds	Die Gesamtzahl der Befehle für die Pub/Sub-Funktionalität. Dies wird aus den OSS Commandstats-Statistiken von Valkey oder Redis abgeleitet, indem alle Befehle zusammengefasst werden, die für die Pub/Sub-Funktionalität verwendet werden: psubscribe, publish, pubsub, punsubscribe, ssubscribe, sunsubscribe, spublish, subscribe und unsubscribe.	Anzahl
PubSubBasedCmdsECPUs	ECPUs wird von Pub/Sub-basierten Befehlen verbraucht.	Anzahl
SetBasedCmds	Gesamtanzahl der Set-basierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Sätze einwirken (scard, sdiff, sadd, sunion usw.).	Anzahl
SetBasedCmdsECPUs	ECPUs wird von mengenbasierten Befehlen verbraucht.	Anzahl

Metrik	Beschreibung	Einheit
SetTypeCmds	Gesamtanzahl der Schreibbe fehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle mutativen Befehlstypen summiert werden, die mit Daten arbeiten (set, hset, sadd, lpop usw.).	Anzahl
SetTypeCmdsECPUs	ECPUs wird von Schreibbe fehlen verbraucht.	Anzahl
SortedSetBasedCmds	Gesamtanzahl der Sorted Set-basierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf eine oder mehrere sortierte Mengen wirken (zcount, zrange, zrank, zadd usw.).	Anzahl
SortedSetBasedCmdsECPUs	ECPUs wird von sortierten Befehlen verbraucht.	Anzahl
StringBasedCmds	Gesamtanzahl der Zeichenfo lge-basierten Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf eine oder mehrere Zeichenketten einwirken (strlen, setex, setrange usw.).	Anzahl

Metrik	Beschreibung	Einheit
StringBasedCmdsECPUs	ECPUs wird von auf Zeichenketten basierenden Befehlen verbraucht.	Anzahl
StreamBasedCmds	Die Gesamtanzahl Stream-basierter Befehle. Dies wird aus der OSS Commandstats-Statistik von Valkey oder Redis abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Stream-Datentypen (xrange, xlen, xadd, xdel usw.) einwirken.	Anzahl
StreamBasedCmdsECPUs	ECPUs wird von stream-basierten Befehlen verbraucht.	Anzahl

## Serverless-Cache-Ereignisse

ElastiCache protokolliert Ereignisse, die sich auf Ihren serverlosen Cache beziehen. Diese Informationen beinhalten Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mithilfe der ElastiCache Konsole, des Befehls AWS CLI `describe-events` oder der Aktion ganz einfach aus dem Protokoll abrufen. ElastiCache API `DescribeEvents`

Sie können wählen, ob Sie ElastiCache Ereignisse mithilfe von Amazon EventBridge überwachen, aufnehmen, transformieren und darauf reagieren möchten. Weitere Informationen finden Sie im Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/Latest/Userguide/>.

### Ereignisse anzeigen (Konsole) ElastiCache

So zeigen Sie Ereignisse mit der ElastiCache Konsole an:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>

2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).
3. Auf dem Bildschirm Ereignisse steht jede Zeile der Liste für ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp, die GMT Uhrzeit des Ereignisses und eine Beschreibung des Ereignisses an. Mit der Option Filter können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

### ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden bis zu 40 Serverless-Cache-Ereignisse aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) für Serverless-Caches aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

### Serverless-Ereignisse

In diesem Abschnitt werden die verschiedenen Arten von Ereignissen dokumentiert, die Sie möglicherweise für Ihre Serverless-Caches erhalten.

#### Ereignisse bei der Erstellung eines Serverless-Caches

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache wurde erstellt	Cache arn	Erstellung	Serverless-Cache	Der Cache <cache-name> wurde erstellt und ist einsatzbereit.
Cache wurde erstellt	Cache arn	Erstellung	Serverless-Cache	Der Cache <cache-name>

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
	Snapshot arn			wird erstellt und die Daten wurden aus dem Snapshot wiederhergestellt. Ihr Cache ist einsatzbereit.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Nicht genügend freie IP-Adressen, um einen Endpunkt zu erstellenVPC.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. In der Anfrage wurden ungültige Subnetze angegeben.

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Das Kontingen tlimit für die Erstellung eines VPC Endpunkts wurde erreicht.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Sie sind nicht berechtigt, einen VPC Endpunkt zu erstellen.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Ein Benutzer mit einer inkompati blen Valkey- oder OSS Redis-Version ist in der Benutzergruppe < > user-group-name vorhanden .

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn Cache snapshot arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Die angegebene Benutzergruppe < user-group-name > existiert nicht.

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	<p>Der Cache &lt;cache-name&gt; konnte nicht erstellt werden. Die Datenwiederherstellung aus dem Snapshot ist fehlgeschlagen, weil &lt;reason&gt;.</p> <p>Gründe für das Fehlschlagen:</p> <ul style="list-style-type: none"> <li>• Die Datei konnte nicht aus S3 abgerufen werden.</li> <li>• Das erwartete md5 stimmt nicht mit dem tatsächlichen md5 überein.</li> <li>• Die bereitgestellte RDB Datei hat eine nicht unterstützte Version.</li> </ul>

## Serverlose Cache-Aktualisierungsereignisse (Valkey oder Redis) OSS

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	SecurityGroups <cache-name> für den Cache aktualisiert.
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Tags für den Cache <cache-name> wurden aktualisiert.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. Ein Benutzer mit einer inkompatiblen Valkey- oder OSS Redis-Version ist in der Benutzergruppe < > user-group-name vorhanden.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	<cache-name> Eine Aktualisierung des Caches ist fehlgeschlagen. SecurityGroups Die

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
				Aktualisierung ist fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist aufgrund unzureichender Berechtigungen fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist fehlgeschlagen, weil SecurityGroups sie ungültig sind.

## Ereignisse zum Löschen von Cache ohne Server (Valkey oder Redis) OSS

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde gelöscht	Cache arn	Löschung	Serverless-Cache	Der Cache <cache-name> wurde gelöscht.

### Ereignisse zur Begrenzung der Cache-Nutzung ohne Server (Valkey oder Redis) OSS

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Limits für den Cache <cache-name> wurden aktualisiert.
Cache-Limit nähert sich	Cache arn	Benachrichtigung	Serverless-Cache	Der Slot <X> verwendet mehr als <Y> % des Limits pro Slot von 32 GB. Der Slot 10 verwendet beispielsweise mehr als 90 % des Limits pro Slot von 32 GB.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist fehlgeschlagen, da der

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
				Cache gelöscht wurde.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist aufgrund einer ungültigen Konfiguration fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist fehlgeschlagen, da die aktuell zwischengespeicherten Daten die neuen Limits überschreiten. Löschen Sie einige Daten, bevor Sie die Limits anwenden.

## Serverlose Cache-Snapshot-Ereignisse (Valkey oder Redis) OSS

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Snapshot wurde erstellt	Cache arn Snapshot arn	Erstellung	serverless-cache-snapshot	Der Snapshot <snapshot-name> wurde für den Cache <cache-name> erstellt.
Snapshot-Erstellung fehlgeschlagen	Cache arn Snapshot arn	Ausfall	serverless-cache-snapshot	<p>Der Snapshot für den Cache &lt;cache-name&gt; konnte nicht erstellt werden. Die Erstellung des Snapshots &lt;snapshot-name&gt; ist mit dem kundenverwalteten Schlüssel &lt;key-id&gt; fehlgeschlagen &lt;reason&gt;.</p> <p>Meldungen zur Fehlerursache:</p> <ul style="list-style-type: none"> <li>• Der kundenverwaltete Schlüssel wurde deaktiviert.</li> <li>• Der kundenverwaltete Schlüssel konnte nicht</li> </ul>

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
				<p>gefunden werden.</p> <ul style="list-style-type: none"> <li>• Das Zeitlimit für die Anfrage wurde überschritten.</li> </ul>
<p>Snapshot-Erstellung fehlgeschlagen</p>	<p>Cache arn Snapshot arn</p>	<p>Ausfall</p>	<p>serverless-cache-snapshot</p>	<p>Der Snapshot für den Cache &lt;cache-name&gt; konnte nicht erstellt werden. Die Erstellung des Snapshots &lt;snapshot-name&gt; ist fehlgeschlagen, weil &lt;reason&gt;.</p> <p>Standardursache:</p> <ul style="list-style-type: none"> <li>• Es liegt ein interner Fehler vor.</li> </ul>

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket %s exportiert werden, da ElastiCache er keine Berechtigungen für den Bucket besitzt.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich bereits ein Objekt mit demselben Namen im Bucket befindet.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich die Konto-ID des Bucket-Eigentümers geändert hat.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den S3-Bucket nicht zugegriffen werden kann.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den Bucket nicht zugegriffen werden kann.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da der Bucket nicht existiert.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte mit dem kundenverwalteten Quell-Snapshot-Schlüssel %s nicht in den Bucket '%s' exportiert werden <reason>.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Quell-Snapshot-Schlüssel <key-id> kopiert werden <reason-name>.
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Ziel-Snapshot-Schlüssel '%s' '%s' kopiert werden.

# Selbst entworfene Cluster-Metriken und Ereignisse für Valkey und Redis OSS

ElastiCache bietet eine Vielzahl von Metriken und Ereignissen für die Überwachung selbst entworfener Cluster bei der Arbeit mit Valkey und Redis. OSS Dazu gehören Metriken auf Hostebene, Metriken auf Befehlsebene und Ereignisprotokolle, die über den AWS CLI und den Amazon Simple Notification Service (SNS) verfügbar sind.

## Themen

- [Metriken für selbst entworfene Cluster](#)
- [Veranstaltungen für selbst entworfene Cluster \(Valkey und Redis\) OSS](#)

## Metriken für selbst entworfene Cluster

Wenn Sie Cluster selbst entwerfen, werden Metriken auf jeder Knotenebene ElastiCache ausgegeben, einschließlich Metriken auf Host-Ebene und Cache-Metriken.

Weitere Informationen zu Metriken auf Host-Ebene finden Sie unter [Metriken auf Host-Ebene](#).

Weitere Informationen zu Metriken auf Knotenebene finden Sie unter [Metriken für Valkey und Redis OSS](#).

## Veranstaltungen für selbst entworfene Cluster (Valkey und Redis) OSS

ElastiCache protokolliert Ereignisse, die sich auf Ihre selbst entworfenen Caches beziehen. Wenn Sie mit selbst entworfenen Clustern arbeiten, können Sie Ihre Cluster-Ereignisse in der ElastiCache Konsole, mithilfe von oder mithilfe von Amazon Simple Notification Service (SNS) anzeigen. AWS CLI Selbst entworfene Cluster-Ereignisse werden nicht auf Amazon EventBridge veröffentlicht.

Informationen über Ereignisse für selbst entworfene Cluster umfassen Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mithilfe der ElastiCache Konsole, des Befehls AWS CLI describe-events oder der Aktion ganz einfach aus dem Protokoll abrufen. ElastiCache API DescribeEvents

### ElastiCache Ereignisse anzeigen (Konsole)

Das folgende Verfahren zeigt Ereignisse mithilfe der ElastiCache Konsole an.

Um Ereignisse mit der ElastiCache Konsole anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).
3. Auf dem Bildschirm Ereignisse steht jede Zeile der Liste für ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp, die GMT Uhrzeit des Ereignisses und eine Beschreibung des Ereignisses an. Mit der Option Filter können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

### ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden bis zu 40 Ereignisse für selbst entworfene Cluster aufgelistet.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) für selbst entworfene Caches aufgelistet.

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

### Ereignisse für selbst entworfene Cluster

Dieser Abschnitt enthält die Liste der Ereignisse, die Sie für Ihre selbst entworfenen Cluster erwarten können.

Die folgenden ElastiCache Ereignisse lösen SNS Amazon-Benachrichtigungen aus. Weitere Informationen zu Ereignisdetails finden Sie unter [ElastiCache Ereignisse anzeigen](#).

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Ein Cache-Knoten wurde zum Cache-Cluster hinzugefügt und steht zur Verwendung bereit.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache: AddCacheNodeFailed aufgrund unzureichender freier IP-Adressen	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Ein Cache-Knoten konnte nicht hinzugefügt werden, da nicht genug IP-Adressen verfügbar sind.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Es wurde mindestens ein Parameter des Cache-Clusters geändert.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	Die Bereitstellung eines Cache-Clusters wurde abgeschlossen und die Cache-Knoten im Cache-Cluster stehen zur Verwendung bereit.
ElastiCache: CacheClusterProvisioningFailed aufgrund eines inkompatiblen Netzwerkzustands	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	Es wurde versucht, einen neuen Cache-Cluster in einer nicht existierenden virtuellen privaten Cloud (VPC) zu starten.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Die Skalierung für den Cache-Cluster wurde erfolgreich abgeschlossen.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	Die Erweiterung des Cache-Clusters ist fehlgeschlagen.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	<p>Eines der folgenden Ereignisse ist aufgetreten:</p> <ul style="list-style-type: none"><li>• Die Liste der Cache-Sicherheitsgruppen, die für den Cache-Cluster autorisiert sind, wurde geändert.</li><li>• Eine oder mehrere neue EC2 Sicherheitsgruppen wurden für eine der Cache-Sicherheitsgruppen autorisiert, die dem Cache-Cluster zugeordnet sind.</li><li>• Eine oder mehrere EC2 Sicherheitsgruppen wurden aus einer der Cache-Sicherheitsgruppen, die dem Cache-Cluster zugeordnet sind, gesperrt.</li></ul>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache hat festgestellt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, heruntergestuft oder nicht erreichbar ist, und hat begonnen, den Cache-Knoten zu ersetzen.</p> <div data-bbox="1068 590 1508 905" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Der DNS Eintrag für den ersetzten Cache-Knoten wurde nicht geändert.</p></div> <p>In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Cli-ent-Bibliotheken verwenden den Cache-Knoten jedoch möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache hat festgestellt, dass der Host, auf dem ein Cacheknoten ausgeführt wird, heruntergefahren oder nicht erreichbar ist, und hat den Austausch des Cacheknotens abgeschlossen.</p> <div data-bbox="1068 590 1507 905"><p> <b>Note</b></p><p>Der DNS Eintrag für den ersetzten Cache-Knoten wurde nicht geändert.</p></div> <p>In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Cli-ent-Bibliotheken verwenden den Cache-Knoten jedoch möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt ElastiCache wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Mindestens ein Cache-Knoten wurde neu gestartet.  Nachricht (Memcached): "Cache node %s shutdown" Dann eine zweite Nachricht: "Cache node %s restarted"
ElastiCache:CertificateRenewalComplete (Nur Valkey oder OSS Redis)	ElastiCache:CertificateRenewalComplete	Das Amazon CA-Zertifikat wurde erfolgreich erneuert.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	Die Replikationsgruppe wurde erfolgreich erstellt.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	Ein Cache-Cluster und alle zugeordneten Cache-Knoten wurden gelöscht.
ElastiCache:FailoverComplete (Nur Valkey oder Redis) OSS	ElastiCache:FailoverComplete : <i>mycluster</i>	Failover zu einem Replikationsknoten wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Die Anzahl der Replikate im Cluster wurde erhöht.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Der Vorgang für das Hinzufügen von Replikaten zu Ihrem Cluster hat begonnen.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster, der ersetzt werden sollte, soll nicht länger ersetzt werden.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>Für einen zu ersetzenden Knoten in Ihrem Cluster wurde eine spätere Ersetzung im neuen Fenster geplant, das in der Benachrichtigung angegeben wurde.</p> <p>Weitere Informationen zu den möglichen Aktionen erhalten Sie unter <a href="#">Knoten ersetzen (Valkey und OSS Redis)</a>.</p>
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	<p>Ein Knoten in Ihrem Cluster soll während des in der Benachrichtigung beschriebenen Fensters ersetzt werden.</p> <p>Weitere Informationen zu den möglichen Aktionen erhalten Sie unter <a href="#">Knoten ersetzen (Valkey und OSS Redis)</a>.</p>
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Ein Cache-Knoten wurde aus dem Cache-Cluster entfernt.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	Die Aufskalierung der Replikationsgruppe wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	Die Aufskalierung der Replikationsgruppe ist fehlgeschlagen.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Für den Knoten ist ein Self-Service-Update verfügbar.
ElastiCache: SnapshotComplete (Nur Valkey oder Redis) OSS	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Ein Cache-Snapshot wurde erfolgreich abgeschlossen.
ElastiCache: SnapshotFailed (Nur Valkey oder Redis) OSS	SnapshotFailed : <i>cluster-name</i>	Ein Cache-Snapshot ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Cache-Ereignissen des Clusters.  Wenn Sie den Snapshot beschreiben (siehe <a href="#">DescribeSnapshots</a> ), ist dessen Status failed.

# Serverlose Metriken und Ereignisse für Memcached

In diesem Abschnitt werden Metriken und Ereignisse beschrieben, die Sie bei der Arbeit mit Serverless-Caches überwachen können.

Themen

- [Metriken für Serverless-Caches](#)
- [Serverless-Cache-Ereignisse](#)

## Metriken für Serverless-Caches

In diesem Abschnitt werden die Metriken und Ereignisse beschrieben, die Sie bei der Arbeit mit serverlosen Memcached-Caches überwachen können.

Der AWS/ElastiCache Namespace umfasst die folgenden CloudWatch Metriken für Ihre serverlosen Memcached-Caches.

Metrik	Beschreibung	Einheit
BytesUsedForCache	Die Gesamtzahl der von den in Ihrem Cache gespeicherten Daten verwendeten Bytes.	Bytes
ElastiCacheProcessingUnits	Die Gesamtzahl der ElastiCacheProcessingUnits (ECPUs), die von den in Ihrem Cache ausgeführten Anfragen verbraucht wurden	Anzahl
SuccessfulReadRequestLatency	Latenz erfolgreicher Leseanfragen.	Mikrosekunden
SuccessfulWriteRequestLatency	Latenz erfolgreicher Schreibanforderungen	Mikrosekunden
TotalCmdsCount	Gesamtzahl aller Befehle, die in Ihrem Cache ausgeführt wurden	Anzahl

Metrik	Beschreibung	Einheit
CurrConnections	Die Anzahl der Client-Verbindungen mit Ihrem Cache.	Anzahl
ThrottledCmds	Die Anzahl der Anfragen, um die gedrosselt wurden, ElastiCache weil die Arbeitslast schneller skaliert wurde, als sie skaliert ElastiCache werden kann.	Anzahl
NewConnections	Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind.	Anzahl
CurrItems	Anzahl der Elemente im Cache.	Anzahl
NetworkBytesIn	Gesamtzahl der in den Cache übertragenen Bytes	Bytes
NetworkBytesOut	Gesamtzahl der aus dem Cache übertragenen Bytes	Bytes
Evictions	Die Anzahl der Schlüssel, die vom Cache bereinigt wurden	Anzahl
Zurückgefordert	Die Anzahl der Schlüssel, die im Cache abgelaufen sind	Anzahl

## Metriken auf Befehlsebene

ElastiCache gibt außerdem die folgenden Memcached-Metriken auf Befehlsebene aus

Metrik	Beschreibung	Einheit
CmdGet	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
CmdSet	Die Anzahl der Set-Befehle, die der Cache empfangen hat.	Anzahl
CmdTouch	Die Anzahl der touch-Befehle, die der Cache erhalten hat.	Anzahl
GetHits	Die Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
GetMisses	Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrHits	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrMisses	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl

Metrik	Beschreibung	Einheit
DecrHits	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DecrMisses	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
DeleteHits	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DeleteMisses	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
TouchHits	Die Anzahl der Schlüssel, auf die zugegriffen wurde und die mit einer neuen Ablaufzeit versehen wurden.	Anzahl
TouchMisses	Die Anzahl der Schlüssel, auf die zwar zugegriffen wurde, die aber nicht gefunden werden konnten.	Anzahl

Metrik	Beschreibung	Einheit
CasHits	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel gefunden wurde und die CAS-Werte übereinstimmen.	Anzahl
CasMisses	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
CasBadval	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der CAS-Wert nicht mit dem gespeicherten CAS-Wert übereinstimmt.	Anzahl
CmdFlush	Anzahl der flush-Befehle, die der Cache erhalten hat.	Anzahl

## Serverless-Cache-Ereignisse

ElastiCache protokolliert Ereignisse, die sich auf Ihren serverlosen Cache beziehen. Diese Informationen beinhalten Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mithilfe der ElastiCache Konsole, des Befehls AWS CLI `describe-events` oder der Aktion ganz einfach aus dem Protokoll abrufen. ElastiCache API `DescribeEvents`

Sie können wählen, ob Sie ElastiCache Ereignisse mithilfe von Amazon EventBridge überwachen, aufnehmen, transformieren und darauf reagieren möchten. Weitere Informationen finden Sie im Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/Latest/Userguide/>.

## Ereignisse anzeigen (Konsole) ElastiCache

So zeigen Sie Ereignisse mit der ElastiCache Konsole an:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).
3. Auf dem Bildschirm Ereignisse steht jede Zeile der Liste für ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp, die GMT Uhrzeit des Ereignisses und eine Beschreibung des Ereignisses an. Mit der Option Filter können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

## ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden bis zu 40 Serverless-Cache-Ereignisse aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) für Serverless-Caches aufgelistet.

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

## Serverless-Ereignisse

In diesem Abschnitt werden die verschiedenen Arten von Ereignissen dokumentiert, die Sie möglicherweise für Ihre Serverless-Caches erhalten.

### Ereignisse bei der Erstellung eines Serverless-Caches

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache wurde erstellt	Cache arn	Erstellung	Serverless-Cache	Der Cache <cache-name>

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
				wurde erstellt und ist einsatzbereit.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Nicht genügend freie IP-Adressen, um einen Endpunkt zu erstellenVPC.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. In der Anfrage wurden ungültige Subnetze angegeben.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Das Kontingentlimit für die Erstellung eines VPC Endpunkts wurde erreicht.

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Sie sind nicht berechtigt, einen VPC Endpunkt zu erstellen.

### Serverlose Cache-Aktualisierungsereignisse (Memcached)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	SecurityGroups <cache-name> für den Cache aktualisiert.
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Tags für den Cache <cache-name> wurden aktualisiert.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist fehlgeschlagen.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name>ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist aufgrund unzureichender Berechtigungen fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name>ist fehlgeschlagen. SecurityGroups Die Aktualisierung ist fehlgeschlagen, weil SecurityGroups sie ungültig sind.

### Ereignisse zum Löschen von Cache ohne Server (Memcached)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Cache wurde gelöscht	Cache arn	Löschung	Serverless-Cache	Der Cache <cache-name> wurde gelöscht.

### Ereignisse zur Begrenzung der Cache-Nutzung ohne Server (Memcached)

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache wurde aktualisiert	Cache arn	Konfigurationsänderung	Serverless-Cache	Die Limits für den Cache <cache-name> wurden aktualisiert.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist fehlgeschlagen, da der Cache gelöscht wurde.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist aufgrund einer ungültigen Konfiguration fehlgeschlagen.

### Serverlose Cache-Snapshot-Ereignisse (Memcached)

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Snapshot wurde erstellt	Cache arn Snapshot arn	Erstellung	serverless-cache-snapshot	Der Snapshot <snapshot-name> wurde für den Cache <cache-name> erstellt.
Snapshot-Erstellung fehlgeschlagen	Cache arn Snapshot arn	Ausfall	serverless-cache-snapshot	<p>Der Snapshot für den Cache &lt;cache-name&gt; konnte nicht erstellt werden. Die Erstellung des Snapshots &lt;snapshot-name&gt; ist mit dem kundenverwalteten Schlüssel &lt;key-id&gt; fehlgeschlagen &lt;reason&gt;.</p> <p>Meldungen zur Fehlerursache:</p> <ul style="list-style-type: none"> <li>• Der kundenverwaltete Schlüssel wurde deaktiviert.</li> <li>• Der kundenverwaltete Schlüssel konnte nicht</li> </ul>

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
				<p>gefunden werden.</p> <ul style="list-style-type: none"> <li>• Das Zeitlimit für die Anfrage wurde überschritten.</li> </ul>
Snapshot-Erstellung fehlgeschlagen	<p>Cache arn</p> <p>Snapshot arn</p>	Ausfall	serverless-cache-snapshot	<p>Der Snapshot für den Cache &lt;cache-name&gt; konnte nicht erstellt werden. Die Erstellung des Snapshots &lt;snapshot-name&gt; ist fehlgeschlagen, weil &lt;reason&gt;.</p> <p>Standardursache:</p> <ul style="list-style-type: none"> <li>• Es liegt ein interner Fehler vor.</li> </ul>

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket %s exportiert werden, da ElastiCache er keine Berechtigungen für den Bucket besitzt.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich bereits ein Objekt mit demselben Namen im Bucket befindet.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da sich die Konto-ID des Bucket-Eigentümers geändert hat.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den S3-Bucket nicht zugegriffen werden kann.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da auf den Bucket nicht zugegriffen werden kann.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden, da der Bucket nicht existiert.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte mit dem kundenverwalteten Quell-Snapshot-Schlüssel %s nicht in den Bucket '%s' exportiert werden <reason>.
Der Export des Snapshots ist fehlgeschlagen.	Snapshot arn	Ausfall	serverless-cache-snapshot	Der Snapshot für den Cache <cache-name> konnte nicht exportiert werden. Der Snapshot konnte nicht in den Bucket '%s' exportiert werden.

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Quell-Snapshot-Schlüssel <key-id> kopiert werden <reason-name>.
Der Snapshot konnte nicht kopiert werden	Snapshot arn-1 Snapshot arn-2	Ausfall	serverless-cache-snapshot	Der Snapshot <snapshot-name> konnte nicht kopiert werden. Der Snapshot '%s' konnte nicht in den Snapshot '%s' mit dem kundenverwalteten Ziel-Snapshot-Schlüssel '%s' '%s' kopiert werden.

# ElastiCache API Amazon-Anrufe protokollieren mit AWS CloudTrail

Amazon ElastiCache ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service bei Amazon ausgeführt wurden ElastiCache. CloudTrail erfasst alle API Aufrufe für Amazon ElastiCache als Ereignisse, einschließlich Anrufe von der ElastiCache Amazon-Konsole und von Codeaufrufen an den ElastiCache API Amazon-Betrieb. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Amazon ElastiCache. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon gestellt wurde ElastiCache, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

## ElastiCache Amazon-Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn in Amazon Aktivitäten auftreten ElastiCache, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich Veranstaltungen für Amazon ElastiCache, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser standardmäßig für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittle die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Unterstützte Dienste und Integrationen](#)
- [Konfiguration von SNS Amazon-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Alle ElastiCache Amazon-Aktionen werden von der Referenz protokolliert CloudTrail und sind in der [ElastiCache APIReferenz](#) dokumentiert. Beispielsweise generieren Aufrufe von DescribeCacheCluster und ModifyCacheCluster Aktionen Einträge in den CloudTrail Protokolldateien. CreateCacheCluster

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder IAM Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie im [CloudTrail userIdentityElement](#).

## ElastiCache Amazon-Protokolldateieinträge verstehen

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die CreateCacheCluster Aktion demonstriert.

```
{
 "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"
}
```

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `DescribeCacheCluster` Aktion demonstriert. Beachten Sie, dass bei allen Amazon ElastiCache Describe-Aufrufen (`Describe*`) der `ResponseElements` Abschnitt entfernt wird und als `null` angezeigt wird.

```
{
 "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"
}
```

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der eine `ModifyCacheCluster` Aktion aufzeichnet.

```
{
 "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"
}
```

## SNSÜberwachung von ElastiCache Ereignissen durch Amazon

Wenn für einen Cluster wichtige Ereignisse eintreten, ElastiCache sendet eine Benachrichtigung an ein bestimmtes SNS Amazon-Thema. Zu den wichtigen Ereignissen zählen beispielsweise das fehlgeschlagene Hinzufügen eines Knotens, das erfolgreiche Hinzufügen eines Knotens und die Änderung einer Sicherheitsgruppe. Durch die Überwachung wichtiger Schlüsselereignisse können Sie den aktuellen Status Ihrer Cluster erfahren und, je nach Ereignis, Korrekturen vornehmen.

### Themen

- [ElastiCache SNSAmazon-Benachrichtigungen verwalten](#)
- [ElastiCache Ereignisse anzeigen](#)
- [Event-Benachrichtigungen und Amazon SNS](#)

## ElastiCache SNSAmazon-Benachrichtigungen verwalten

Mit Amazon Simple Notification Service (AmazonSNS) können Sie konfigurieren ElastiCache , dass Benachrichtigungen für wichtige Cluster-Ereignisse gesendet werden. In diesen Beispielen konfigurieren Sie einen Cluster mit dem Amazon-Ressourcennamen (ARN) eines SNS Amazon-Themas für den Empfang von Benachrichtigungen.

### Note

- In diesem Thema wird davon ausgegangen, dass Sie sich bei Amazon angemeldet SNS und ein SNS Amazon-Thema eingerichtet und abonniert haben. Informationen dazu finden Sie im [Entwicklerhandbuch zu Amazon Simple Notification Service](#).
- Standardmäßig API `modify-replication-group` betrifft das alle Gruppen in einer Region und nicht nur die aktuell angegebene Gruppe. Wenn Sie eine bestimmte Gruppe in einer Region anders konfigurieren möchten als die anderen Gruppen, können Sie die `--notification-topic-arn` Option verwenden, um ein separates Thema für diese Gruppe zu erstellen.

## Ein SNS Amazon-Thema hinzufügen

In den folgenden Abschnitten erfahren Sie, wie Sie mithilfe der AWS Konsole, der oder der AWS CLI ein SNS Amazon-Thema hinzufügen ElastiCache API.

### Ein SNS Amazon-Thema hinzufügen (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie ein SNS Amazon-Thema für einen Cluster hinzufügen. Wenn Sie Valkey oder Redis verwenden OSS, um in Schritt 2 ein SNS Amazon-Thema für eine Replikationsgruppe hinzuzufügen, wählen Sie statt eines Clusters eine Replikationsgruppe aus. Folgen Sie dann den verbleibenden Schritten.

#### Note

Dieser Prozess kann auch verwendet werden, um das SNS Amazon-Thema zu ändern.

Um ein SNS Amazon-Thema für einen Cluster hinzuzufügen oder zu ändern (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie unter Cluster den Cluster aus, für den Sie ein SNS Amazon-Thema hinzufügen oder ändern möchten ARN.
3. Wählen Sie Ändern aus.
4. Wählen Sie unter Cluster modifizieren unter Thema für SNS Benachrichtigung das SNS Thema aus, das Sie hinzufügen möchten, oder wählen Sie Manuelle ARN Eingabe und geben Sie das ARN SNS Amazon-Thema ein.
5. Wählen Sie Ändern aus.

### Ein SNS Amazon-Thema hinzufügen (AWS CLI)

Verwenden Sie den AWS CLI Befehl, um ein SNS Amazon-Thema für einen Cluster hinzuzufügen oder zu ändern `modify-cache-cluster`.

Das folgende Codebeispiel fügt ein SNS Amazon-Thema arn zu my-cluster hinzu.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \
```

```
--cache-cluster-id my-cluster \
--notification-topic-arn arn:aws:sns:us-
west-2:123456789xxx:ElastiCacheNotifications
```

Für Windows:

```
aws elasticache modify-cache-cluster ^
--cache-cluster-id my-cluster ^
--notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

Weitere Informationen finden Sie unter [modify-cache-cluster](#).

Ein SNS Amazon-Thema hinzufügen (ElastiCache API)

Um ein SNS Amazon-Thema für einen Cluster hinzuzufügen oder zu ändern, rufen Sie die `ModifyCacheCluster` Aktion mit den folgenden Parametern auf:

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/
?Action=ModifyCacheCluster
&ApplyImmediately=false
&CacheClusterId=my-cluster
&NotificationTopicArn=arn%3Aaws%3Asns%3Aus-
west-2%3A565419523791%3AElastiCacheNotifications
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter [ModifyCacheCluster](#).

## SNSAmazon-Benachrichtigungen aktivieren und deaktivieren

Sie können Benachrichtigungen für einen Cluster aktivieren oder deaktivieren. Die folgenden Verfahren zeigen Ihnen, wie Sie SNS Amazon-Benachrichtigungen deaktivieren können.

### SNSAmazon-Benachrichtigungen aktivieren und deaktivieren (Konsole)

Um SNS Amazon-Benachrichtigungen zu deaktivieren, verwenden Sie den AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste der Cluster anzuzeigen, die Memcached ausführen, wählen Sie im Navigationsbereich Memcached.

Um eine Liste Ihrer Cluster anzuzeigen, auf denen Valkey oder Redis ausgeführt werdenOSS, wählen Sie im Navigationsbereich Valkey oder Redis aus. OSS

3. Aktivieren Sie das Kontrollkästchen links neben dem Cluster, für den die Benachrichtigung geändert werden soll.
4. Wählen Sie Ändern aus.
5. Wählen Sie unter „Cluster modifizieren“ unter „Thema für SNS Benachrichtigungen“ die Option Benachrichtigungen deaktivieren aus.
6. Wählen Sie Ändern aus.

### SNSAmazon-Benachrichtigungen aktivieren und deaktivieren (AWS CLI)

Verwenden Sie den Befehl `modify-cache-cluster` mit den folgenden Parametern, um SNS Amazon-Benachrichtigungen zu deaktivieren:

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-cluster \
 --notification-topic-status inactive
```

Für Windows:

```
aws elasticache modify-cache-cluster ^
```

```
--cache-cluster-id my-cluster ^
--notification-topic-status inactive
```

## SNS Amazon-Benachrichtigungen aktivieren und deaktivieren () ElastiCache API

Um SNS Amazon-Benachrichtigungen zu deaktivieren, rufen Sie die `ModifyCacheCluster` Aktion mit den folgenden Parametern auf:

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Diese Aktion führt zu folgender oder einer ähnlichen Ausgabe:

### 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>
```

## ElastiCache Ereignisse anzeigen

ElastiCache protokolliert Ereignisse, die sich auf Ihre Cluster-Instances, Sicherheitsgruppen und Parametergruppen beziehen. Diese Informationen beinhalten Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mithilfe der ElastiCache Konsole, des AWS CLI `describe-events` Befehls oder der ElastiCache API Aktion `DescribeEvents` ganz einfach aus dem Protokoll abrufen.

Die folgenden Verfahren zeigen Ihnen, wie Sie alle ElastiCache Ereignisse der letzten 24 Stunden (1440 Minuten) anzeigen können.

### ElastiCache Ereignisse anzeigen (Konsole)

Das folgende Verfahren zeigt Ereignisse mithilfe der ElastiCache Konsole an.

Um Ereignisse mit der ElastiCache Konsole anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).

Auf dem Bildschirm Ereignisse steht jede Zeile der Liste für ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp (Cache-Cluster, cache-parameter-group, cache-security-group, oder cache-subnet-group), die GMT Uhrzeit des Ereignisses und eine Beschreibung des Ereignisses an.

Mit der Option `Filter` können Sie angeben, ob alle Ereignisse oder nur Ereignisse eines bestimmten Typs in der Ereignisliste enthalten sein sollen.

### ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden bis zu 40 Cache-Cluster-Ereignisse aufgelistet.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) aufgelistet.

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

Die Ausgabe des Befehls `describe-events` sieht in etwa wie folgt aus:

```
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"
 }
]
}
```

Weitere Informationen z. B. zu den verfügbaren Parametern und den zulässigen Parameterwerten finden Sie unter [describe-events](#).

## ElastiCache Ereignisse anzeigen (ElastiCache API)

Verwenden Sie die DescribeEvents Aktion ElastiCache API, um eine Liste von ElastiCache Ereignissen mithilfe von zu generieren. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden die letzten 40 Cache-Cluster-Ereignisse aufgelistet.

```
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>
```

Mit dem folgenden Code werden alle Cache-Cluster-Ereignisse der letzten 24 Stunden (1 440 Minuten) aufgelistet.

```
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>
```

Die Ausgabe der oben angegebenen Aktionen sollte in etwa wie folgt aussehen:

```
<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>
```

Weitere Informationen z. B. zu den verfügbaren Parametern und den zulässigen Parameterwerten finden Sie unter [DescribeEvents](#).

## Event-Benachrichtigungen und Amazon SNS

ElastiCache kann Nachrichten mit Amazon Simple Notification Service (SNS) veröffentlichen, wenn wichtige Ereignisse in einem Cache-Cluster auftreten. Mit dieser Funktion können die Serverlisten auf Client-Computern aktualisiert werden, die mit einzelnen Cache-Knotenendpunkten eines Cache-Clusters verbunden sind.

### Note

Weitere Informationen zu Amazon Simple Notification Service (SNS), einschließlich Preisinformationen und Links zur SNS Amazon-Dokumentation, finden Sie auf der [SNS Amazon-Produktseite](#).

Benachrichtigungen werden zu einem bestimmten SNS Amazon-Thema veröffentlicht. Für Benachrichtigungen müssen folgende Voraussetzungen erfüllt sein:

- Für ElastiCache Benachrichtigungen kann nur ein Thema konfiguriert werden.
- Das AWS Konto, dem das SNS Amazon-Thema gehört, muss dasselbe Konto sein, das den Cache-Cluster besitzt, auf dem Benachrichtigungen aktiviert sind.
- Das SNS Amazon-Thema, in dem Sie veröffentlichen, kann nicht verschlüsselt werden.

### Note

Es ist möglich, ein verschlüsselt (im Ruhezustand) SNS Amazon-Thema an den Cluster anzuhängen. Der Status des Themas in der ElastiCache Konsole wird jedoch als inaktiv angezeigt, wodurch das Thema effektiv vom Cluster getrennt wird, wenn Nachrichten an das ElastiCache Thema weitergeleitet werden.

- Das SNS Amazon-Thema muss sich in derselben Region wie der ElastiCache Cluster befinden.

## ElastiCache Ereignisse

Die folgenden ElastiCache Ereignisse lösen SNS Amazon-Benachrichtigungen aus. Weitere Informationen zu Ereignisdetails finden Sie unter [ElastiCache Ereignisse anzeigen](#).

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	Ein Cache-Knoten wurde zum Cache-Cluster hinzugefügt und steht zur Verwendung bereit.
ElastiCache: AddCacheNodeFailed aufgrund unzureichender freier IP-Adressen	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Ein Cache-Knoten konnte nicht hinzugefügt werden, da nicht genug IP-Adressen verfügbar sind.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Es wurde mindestens ein Parameter des Cache-Clusters geändert.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioning	Die Bereitstellung eines Cache-Clusters wurde abgeschlossen und die

Ereignisname	Fehlermeldung	Beschreibung
	Complete <i>cluster-name-0001-005</i>	Cache-Knoten im Cache-Cluster stehen zur Verwendung bereit.
ElastiCache: CacheClusterProvisioningFailed aufgrund eines inkompatiblen Netzwerkstatus	ElastiCache:CacheClusterProvisioning Failed : <i>cluster-name</i>	Es wurde versucht, einen neuen Cache-Cluster in einer nicht existierenden virtuellen privaten Cloud (VPC) zu starten.
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	Die Skalierung für den Cache-Cluster wurde erfolgreich abgeschlossen.
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed : <i>cluster-name</i>	Die Erweiterung des Cache-Clusters ist fehlgeschlagen.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	<p>Eines der folgenden Ereignisse ist aufgetreten:</p> <ul style="list-style-type: none"><li>• Die Liste der Cache-Sicherheitsgruppen, die für den Cache-Cluster autorisiert sind, wurde geändert.</li><li>• Eine oder mehrere neue EC2 Sicherheitsgruppen wurden für eine der Cache-Sicherheitsgruppen autorisiert, die dem Cache-Cluster zugeordnet sind.</li><li>• Eine oder mehrere EC2 Sicherheitsgruppen wurden aus einer der Cache-Sicherheitsgruppen, die dem Cache-Cluster zugeordnet sind, gesperrt.</li></ul>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache hat festgestellt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, heruntergestuft oder nicht erreichbar ist, und hat begonnen, den Cache-Knoten zu ersetzen.</p> <div data-bbox="1068 590 1507 905" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Der DNS Eintrag für den ersetzten Cache-Knoten wurde nicht geändert.</p></div> <p>In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Client-Bibliotheken verwenden den Cache-Knoten jedoch möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt ElastiCache wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache hat festgestellt, dass der Host, auf dem ein Cacheknoten ausgeführt wird, heruntergefahren oder nicht erreichbar ist, und hat den Austausch des Cacheknotens abgeschlossen.</p> <div data-bbox="1068 590 1507 905" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Der DNS Eintrag für den ersetzten Cache-Knoten wurde nicht geändert.</p></div> <p>In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Client-Bibliotheken verwenden den Cache-Knoten jedoch möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt ElastiCache wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Mindestens ein Cache-Knoten wurde neu gestartet.  Nachricht (Memcache d): "Cache node %s shutdown" Dann eine zweite Nachricht: "Cache node %s restarted"
ElastiCache:CertificateRenewalComplete (Nur Valkey oder OSS Redis)	ElastiCache:CertificateRenewalComplete	Das Amazon CA-Zertifikat wurde erfolgreich erneuert.
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	Die Replikationsgruppe wurde erfolgreich erstellt.
ElastiCache>DeleteCacheClusterComplete	ElastiCache>DeleteCacheClusterComplete : <i>cluster-name</i>	Ein Cache-Cluster und alle zugeordneten Cache-Knoten wurden gelöscht.
ElastiCache:FailoverComplete (Nur Valkey oder Redis) OSS	ElastiCache:FailoverComplete : <i>mycluster</i>	Failover zu einem Replikationsknoten wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Die Anzahl der Replikate im Cluster wurde erhöht.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Der Vorgang für das Hinzufügen von Replikaten zu Ihrem Cluster hat begonnen.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster, der ersetzt werden sollte, soll nicht länger ersetzt werden.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>Für einen zu ersetzenden Knoten in Ihrem Cluster wurde eine spätere Ersetzung im neuen Fenster geplant, das in der Benachrichtigung angegeben wurde.</p> <p>Weitere Informationen zu den möglichen Aktionen erhalten Sie unter <a href="#">Knoten ersetzen (Valkey und OSS Redis)</a>.</p>
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	<p>Ein Knoten in Ihrem Cluster soll während des in der Benachrichtigung beschriebenen Fensters ersetzt werden.</p> <p>Weitere Informationen zu den möglichen Aktionen erhalten Sie unter <a href="#">Knoten ersetzen (Valkey und OSS Redis)</a>.</p>
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Ein Cache-Knoten wurde aus dem Cache-Cluster entfernt.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	Die Aufskalierung der Replikationsgruppe wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	Die Aufskalierung der Replikationsgruppe ist fehlgeschlagen.
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	Für den Knoten ist ein Self-Service-Update verfügbar.
ElastiCache: SnapshotComplete (Nur Valkey oder Redis) OSS	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Ein Cache-Snapshot wurde erfolgreich abgeschlossen.
ElastiCache: SnapshotFailed (Nur Valkey oder Redis) OSS	SnapshotFailed : <i>cluster-name</i>	Ein Cache-Snapshot ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Cache-Ereignissen des Clusters.  Wenn Sie den Snapshot beschreiben (siehe <a href="#">DescribeSnapshots</a> ), ist dessen Status failed.

## Verwandte Themen

- [ElastiCache Ereignisse anzeigen](#)

# Protokollzustellung

## Note

Slow Log wird für Valkey 7.x und höher sowie für OSS Redis-Cache-Cluster und Replikationsgruppen ab der Engine-Version 6.0 unterstützt.

Engine Log wird für Valkey 7.x- und OSS Redis-Cache-Cluster und Replikationsgruppen ab Engine-Version 6.2 unterstützt.

Mit der Protokollzustellung können Sie [SLOWLOG](#) oder Engine Log an eines von zwei Zielen streamen:

- Amazon Data Firehose
- CloudWatch Amazon-Protokolle

Sie aktivieren und konfigurieren die Protokollzustellung, wenn Sie einen Cluster erstellen oder ändern mit ElastiCache APIs. Jeder Protokolleintrag wird in einem von zwei Formaten an das angegebene Ziel übermittelt: JSON oder TEXT.

Eine feste Anzahl von Slow-Protokolleinträgen wird regelmäßig von der Engine abgerufen. Je nach dem für den Engine-Parameter `slowlog-max-len` angegebenen Wert werden zusätzliche langsame Protokolleinträge möglicherweise nicht an das Ziel übermittelt.

Sie können die Versandkonfigurationen ändern oder die Protokollzustellung jederzeit deaktivieren. Verwenden Sie dazu die AWS Konsole oder eine der Optionen zum Ändern APIs, entweder [modify-cache-cluster](#) oder [modify-replication-group](#).

Sie müssen den `apply-immediately`-Parameter für alle Änderungen der Protokollzustellung setzen.

## Note

Amazon CloudWatch Logs-Gebühren fallen an, wenn die Protokollzustellung aktiviert ist, auch wenn Protokolle direkt an Amazon Data Firehose geliefert werden. Weitere Informationen finden Sie im Abschnitt Verkaufte Logs in der [CloudWatch Amazon-Preisübersicht](#).

## Inhalt eines langsamen Protokolleintrags

Das Slow Log enthält die folgenden Informationen:

- `CacheClusterId`— Die ID des Cache-Clusters
- `CacheNodeId`— Die ID des Cache-Knotens
- `Id` – Eine eindeutige progressive Kennung für jeden langsamen Protokoll-Eintrag
- `Timestamp` – Der Unix-Zeitstempel, zu dem der protokollierte Befehl verarbeitet wurde
- `Duration (Dauer)` – Die für die Ausführung benötigte Zeit in Mikrosekunden
- `Command` – Der vom Client verwendete Befehl. Zum Beispiel, `set foo bar` wo `foo` ist der Schlüssel und `bar` ist der Wert. ElastiCache ersetzt den tatsächlichen Schlüsselnamen und -wert durch, `(2 more arguments)` um zu verhindern, dass vertrauliche Daten offengelegt werden.
- `ClientAddress`— IP-Adresse und Port des Clients
- `ClientName`— Clientname, falls er über den `CLIENT SETNAME` Befehl gesetzt wurde

## Inhalte eines Engine-Protokolleintrags

Das ElastiCache Engine-Protokoll enthält die folgenden Informationen:

- `CacheClusterId`— Die ID des Cache-Clusters
- `CacheNodeId`— Die ID des Cache-Knotens
- `Protokollebene` — `LogLevel` kann eine der folgenden Optionen sein: `VERBOSE("-")`, `NOTICE("*")`, `WARNING("#")`.
- `Zeit` — Die UTC Uhrzeit der protokollierten Nachricht. Die Zeit ist in folgendem Format angegeben: `"DD MMM YYYY hh:mm:ss.ms UTC"`
- `Rolle` – Rolle des Knotens, von dem das Protokoll ausgegeben wird. Dabei kann es sich um einen der folgenden Werte handeln: „M“ für Primär, „S“ für Replikat, „C“ für einen untergeordneten Writer-Prozess, der anRDB/arbeitet, AOF oder „X“ für Sentinel.
- `Nachricht` — Engine-Log-Meldung.

## Berechtigungen zum Konfigurieren der Protokollierung

Sie müssen die folgenden IAM Berechtigungen in Ihre IAM Benutzer-/Rollenrichtlinie aufnehmen:

- logs:CreateLogDelivery
- logs:UpdateLogDelivery
- logs>DeleteLogDelivery
- logs:GetLogDelivery
- logs>ListLogDeliveries

Weitere Informationen finden Sie unter [Übersicht über die Zugriffsverwaltung: Berechtigungen und Richtlinien](#).

## Spezifikationen des Protokolltyps und des Protokollformats

### Slow-Protokoll

Slow Log unterstützt sowohl als JSON TEXT

Im Folgenden wird ein JSON Formatbeispiel gezeigt:

```
{
 "CacheClusterId": "logslowxxxxmsxj",
 "CacheNodeId": "0001",
 "Id": 296,
 "Timestamp": 1605631822,
 "Duration (us)": 0,
 "Command": "GET ... (1 more arguments)",
 "ClientAddress": "192.168.12.104:55452",
 "ClientName": "logslowxxxxmsxj##"
}
```

Im Folgenden wird ein TEXT Formatbeispiel gezeigt:

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

### Engine-Protokoll

Das Engine-Protokoll unterstützt JSON sowohl als TEXT

Im Folgenden wird ein JSON Formatbeispiel gezeigt:

```
{
 "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"
}
```

Im Folgenden wird ein TEXT Formatbeispiel gezeigt:

```
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 Ziele protokollieren

In diesem Abschnitt werden die Protokollierungsziele beschrieben, die Sie für Ihre ElastiCache Protokolle auswählen können. Jeder Abschnitt enthält Anleitungen zum Konfigurieren der Protokollierung für den Zieltyp und Informationen zu jedem Verhalten, das für den jeweiligen Zieltyp spezifisch ist. Nachdem Sie Ihr Protokollierungsziel konfiguriert haben, können Sie dessen Spezifikationen für die ElastiCache Protokollierungskonfiguration angeben, um mit der Protokollierung zu beginnen.

### Themen

- [CloudWatch Amazon-Protokolle](#)
- [Amazon Data Firehose](#)

### CloudWatch Amazon-Protokolle

- Sie geben eine CloudWatch Logs-Protokollgruppe an, in die die Protokolle geliefert werden.
- Protokolle von mehreren Valkey- oder OSS Redis-Clustern und Replikationsgruppen können an dieselbe Protokollgruppe übermittelt werden.
- Für jeden Knoten innerhalb eines Cache-Clusters oder einer Replikationsgruppe wird ein neuer Protokolldatenstrom erstellt und die Protokolle werden an die jeweiligen Protokolldatenströme übermittelt. Der Name des Protokolldatenstroms verwendet das folgende Format: `elasticache/{engine-name}/{cache-cluster-id}/{cache-node-id}/{log-type}`

## Berechtigungen zum Veröffentlichen von Protokollen in Logs CloudWatch

Sie müssen über die folgenden Berechtigungseinstellungen verfügen, um das Senden von Protokollen an eine CloudWatch Logs-Protokollgruppe konfigurieren ElastiCache zu können:

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

Weitere Informationen finden Sie unter An [Protokolle gesendete CloudWatch Protokolle](#).

## Amazon Data Firehose

- Sie geben einen Firehose-Lieferstream an, in den die Protokolle geliefert werden.

- Protokolle aus mehreren Valkey- oder OSS Redis-Clustern und Replikationsgruppen können an denselben Delivery Stream übermittelt werden.
- Protokolle von jedem Knoten innerhalb eines Cache-Clusters oder einer Replikationsgruppe werden an denselben Bereitstellungsdatenstrom übermittelt. Sie können Protokollnachrichten von verschiedenen Cache-Knoten basierend auf `cache-cluster-id` und `cache-node-id` in jeder Protokollnachricht unterscheiden.
- Die Lieferung von Baumstämmen an Firehose ist derzeit in der Region Asien-Pazifik (Osaka) nicht verfügbar.

## Berechtigungen zum Veröffentlichen von Protokollen auf Firehose

Sie benötigen die folgenden Berechtigungen, um das Senden von Protokollen an einen Amazon Kinesis Data Firehose Firehose-Lieferstream zu konfigurieren ElastiCache .

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

## Angeben der Protokollzustellung mithilfe der Konsole

Mit dem können AWS Management Console Sie einen Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert) erstellen, indem Sie den Schritten unter folgen, [Erstellen eines Valkey-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#) oder einen Valkey- oder Redis-Cluster OSS (Cluster-Modus aktiviert) mithilfe der Schritte unter erstellen. [Erstellen eines Valkey- oder Redis-Clusters OSS \(Cluster-Modus aktiviert\) \(Konsole\)](#) In beiden Fällen konfigurieren Sie die Protokollzustellung wie folgt;

1. Wählen Sie unter Erweiterte Einstellungen die Option Logs aus und aktivieren Sie dann entweder Slow Logs oder Engine Logs.
2. Wählen Sie unter Protokollformat entweder Text oder aus JSON.
3. Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
4. Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren Amazon S3 S3-Bucket-Namen, CloudWatchLogs Protokollgruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen ein, oder wählen Sie Bestehende auswählen und dann entweder Ihren CloudWatch Logs-Gruppennamen oder Ihren Kinesis Data Firehose Firehose-Streamnamen aus.

Beim Ändern eines Clusters:

Sie können wählen, ob Sie die Protokollzustellung aktivieren/deaktivieren oder den Zieltyp, das Format oder das Ziel ändern möchten:

1. Melden Sie sich bei der Konsole an und öffnen Sie die Konsole unter. ElastiCache <https://console.aws.amazon.com/elasticache/>
2. Wählen Sie im Navigationsbereich Valkey-Cluster oder OSSRedis-Cluster aus.

3. Suchen Sie in der Liste der Cluster den Cluster aus, den Sie ändern möchten. Wählen Sie den Clusternamen und nicht das Kontrollkästchen daneben.
4. Wählen Sie auf der Seite Clusternamen die Registerkarte Protokolle.
5. Um langsame Protokolle zu aktivieren oder zu deaktivieren, wählen Sie entweder Langsame Protokolle aktivieren oder Langsame Protokolle deaktivieren.
6. Um die Engine-Protokolle zu aktivieren/deaktivieren, wählen Sie entweder Enable engine logs (Engine-Protokolle aktivieren) oder Disable engine logs (Engine-Protokolle deaktivieren) aus.
7. Um Ihre Konfiguration zu ändern, wählen Sie entweder Modify slow logs (Langsame Protokolle ändern) oder Modify engine logs (Engine-Protokolle ändern) aus:
  - Wählen Sie unter Zieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
  - Wählen Sie unter Protokollziel entweder Neu erstellen aus und geben Sie entweder Ihren CloudWatchLogs Protokollgruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen ein. Oder wählen Sie Select existing und wählen Sie dann entweder Ihren CloudWatchLogs Log-Gruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen.

## Angeben der Protokollzustellung mit dem AWS CLI

### Slow-Protokoll

Erstellen Sie eine Replikationsgruppe mit langsamer Protokollzustellung an CloudWatch Logs.

Für Linux, macOS oder 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"
 }'
```

### Für 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"
 }'
```

Ändern Sie eine Replikationsgruppe, um das Protokoll langsam in CloudWatch Logs zu speichern

### Für Linux, macOS oder 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"
 }'
```

## Für 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"
}'
```

Ändern einer Replikationsgruppe, um die langsame Protokollzustellung zu deaktivieren

## Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id test-slow-log \
 --apply-immediately \
 --log-delivery-configurations '
{
 "LogType":"slow-log",
 "Enabled":false
}'
```

## Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id test-slow-log ^
 --apply-immediately ^
 --log-delivery-configurations '
{
 "LogType":"slow-log",
 "Enabled":false
}'
```

## Engine-Protokoll

Erstellen Sie eine Replikationsgruppe mit der Übermittlung von CloudWatch Engine-Protokollen an Logs.

Für Linux, macOS oder 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"
 }'
```

Für 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"
 }'
```

Ändern Sie eine Replikationsgruppe, um das Engine-Protokoll an Firehose zu übermitteln

## Für Linux, macOS oder 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"
 }'
```

## Für 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"
 }'
```

## Ändern einer Replikationsgruppe zum Wechseln in das Engine-Format

### Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id test-slow-log \
 --apply-immediately \
 --log-delivery-configurations '
```

```
{
 "LogType":"engine-log",
 "LogFormat":"json"
}'
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id test-slow-log ^
 --apply-immediately ^
 --log-delivery-configurations '
{
 "LogType":"engine-log",
 "LogFormat":"json"
}'
```

Ändern einer Replikationsgruppe zur Deaktivierung der Übermittlung von Engine-Protokollen

Für Linux, macOS oder Unix:

```
aws elasticache modify-replication-group \
 --replication-group-id test-slow-log \
 --apply-immediately \
 --log-delivery-configurations '
{
 "LogType":"engine-log",
 "Enabled":false
}'
```

Für Windows:

```
aws elasticache modify-replication-group ^
 --replication-group-id test-slow-log ^
 --apply-immediately ^
 --log-delivery-configurations '
{
 "LogType":"engine-log",
 "Enabled":false
}'
```

# Überwachung der Nutzung mit CloudWatch Metrics

ElastiCache bietet Metriken, mit denen Sie Ihre Cluster überwachen können. Sie können auf diese Metriken zugreifen über CloudWatch. Weitere Informationen zu CloudWatch finden Sie in der [CloudWatch Dokumentation](#).

ElastiCache bietet sowohl Metriken auf Host-Ebene (z. B. CPU Nutzung) als auch Metriken, die für die Cache-Engine-Software spezifisch sind (z. B. Cache-Abrufe und Cache-Fehlschläge). Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.

## Important

Sie sollten erwägen, CloudWatch Alarmer für bestimmte wichtige Messwerte einzurichten, damit Sie benachrichtigt werden, wenn sich die Leistung Ihres Cache-Clusters zu verschlechtern beginnt. Weitere Informationen finden Sie unter [Welche Metriken sollte ich überwachen?](#) in diesem Handbuch.

## Themen

- [Metriken auf Host-Ebene](#)
- [Metriken für Valkey und Redis OSS](#)
- [Metriken für Memcached](#)
- [Welche Metriken sollte ich überwachen?](#)
- [Auswählen von Metrikstatistiken und -zeiträumen](#)
- [Überwachung von CloudWatch Cluster- und Knotenmetriken](#)

## Metriken auf Host-Ebene

Der AWS/ElastiCache-Namespace enthält die folgenden Metriken auf Host-Ebene für einzelne Cache-Knoten. Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.

Weitere Informationen finden Sie auch unter:

- [Metriken für Valkey und Redis OSS](#)

Metrik	Beschreibung	Einheit
<code>CPUUtilization</code>	Der Prozentsatz der CPU Auslastung für den gesamten Host. Da Valkey und Redis Single-Threading OSS verwenden, empfehlen wir, die <code>EngineCPUUtilization</code> Metrik für Knoten mit 4 oder mehr Knoten zu überwachen. <code>vCPUs</code>	Prozent
<code>CPUCreditBalance</code>	<p>Die Anzahl der verdienten CPU Credits, die eine Instance seit ihrem Start oder Start gesammelt hat. Für T2 Standard beinhaltet dies <code>CPUCreditBalance</code> auch die Anzahl der angesammelten Start-Credits.</p> <p>Guthaben werden auf dem Guthaben-Konto angesammelt, nachdem sie verdient wurden, und davon entfernt, wenn sie verbraucht werden. Der Guthaben-Kontostand hat ein maximales Limit, das anhand der Instance-Größe bestimmt wird. Nachdem das Limit erreicht ist, verfallen alle neu verdienten Guthabepunkte. Für T2 Standard zählen Startguthaben nicht zum Limit.</p> <p>Die Credits in der <code>CPUCreditBalance</code> stehen der Instance zur Verfügung, um sie über die CPU Basisnutzung hinaus auszugeben.</p> <p><code>CPUKreditkennzahlen</code> sind nur in einem Intervall von fünf Minuten verfügbar.</p> <p>Diese Metrik ist nicht für T2-Instances mit Spitzenleistung verfügbar.</p>	Credits (v MinutenCPU)
<code>CPUCreditUsage</code>	Die Anzahl der CPU Credits, die von der Instance für die CPU Nutzung ausgegeben wurden. Ein CPU Guthaben entspricht einem	Credits (v MinutenCPU)

Metrik	Beschreibung	Einheit
	<p>V, das eine Minute lang bei 100% Auslastung CPU läuft vCPUs, oder einer gleichwertigen Kombination aus Auslastung und Zeit (z. B. ein V, das zwei Minuten lang mit 50% Auslastung CPU betrieben wird, oder zwei, die zwei Minuten lang bei 25% Auslastung vCPUs laufen).</p> <p>CPUKreditkennzahlen sind nur in einem Intervall von fünf Minuten verfügbar. Wenn Sie einen Zeitraum von mehr als fünf Minuten angeben, verwenden Sie die Summen-Statistik anstelle der Durchschnitts-Statistik.</p> <p>Diese Metrik ist nicht für T2-Instances mit Spitzenleistung verfügbar.</p>	
FreeableMemory	Größe des freien Arbeitsspeichers auf dem Host. Dies wird aus den Puffern und dem RAM Cache abgeleitet, die vom Betriebssystem als frei verfügbar gemeldet werden.	Bytes
NetworkBytesIn	Anzahl der Byte, die der Host aus dem Netzwerk gelesen hat.	Bytes
NetworkBytesOut	Anzahl der von der Instance auf allen Netzwerkschnittstellen gesendeten Byte.	Bytes
NetworkPacketsIn	Anzahl der von der Instance auf allen Netzwerkschnittstellen empfangenen Pakete. Diese Metrik gibt das an eine einzelne Instance eingehende Netzwerkdatenvolumen an, ausgedrückt in Anzahl an Paketen.	Anzahl

Metrik	Beschreibung	Einheit
NetworkPacketsOut	Anzahl der von der Instance auf allen Netzwerkschnittstellen gesendeten Pakete. Diese Metrik gibt das von einer einzelnen Instance ausgehende Netzwerkdatenvolumen an, ausgedrückt in Anzahl an Paketen.	Anzahl
NetworkBandwidthInAllowanceExceeded	Die Anzahl der Pakete, die in die Warteschlange gestellt oder verworfen wurden, da die eingehende aggregierte Bandbreite das Maximum für die Instance überschritten hat.	Anzahl
NetworkConntrackAllowanceExceeded	Die Anzahl der verworfenen Pakete, weil die Verbindungsverfolgung das Maximum für die Instance überschritten hat und keine neuen Verbindungen hergestellt werden konnten. Dies kann zu einem Paketverlust für den Datenverkehr zur oder von der Instance führen.	Anzahl
NetworkBandwidthOutAllowanceExceeded	Die Anzahl der Pakete, die in die Warteschlange gestellt oder verworfen wurden, weil die ausgehende aggregierte Bandbreite das Maximum für die Instance überschritten hat.	Anzahl
NetworkPacketsPerSecondAllowanceExceeded	Die Anzahl der Pakete, die in Warteschlangen gestellt oder verworfen wurden, weil Anzahl der bidirektionalen Pakete pro Sekunde das Maximum für die Instance überschritten hat.	Anzahl
NetworkMaxBytesIn	Die maximale Anzahl an empfangenen Byte pro Sekunde pro Sekunde pro Minute.	Bytes
NetworkMaxBytesOut	Die maximale Anzahl an übertragenen Byte pro Sekunde pro Sekunde pro Minute.	Bytes
NetworkMaxPacketsIn	Die maximale Anzahl pro Sekunde empfangener Pakete pro Sekunde pro Minute.	Anzahl

Metrik	Beschreibung	Einheit
NetworkMaxPacketsOut	Die maximale Anzahl an übertragenen Paketen pro Sekunde pro Sekunde pro Minute.	Anzahl
SwapUsage	Größe des belegten Auslagerungsspeichers auf dem Host.	Bytes

## Metriken für Valkey und Redis OSS

Der Amazon ElastiCache Namespace umfasst die folgenden Valkey- und Redis-Metriken. OSS Diese Metriken sind identisch, wenn die Valkey-Engine verwendet wird.

Mit Ausnahme von ReplicationLag und EngineCPUUtilization werden diese Metriken aus dem info Befehl abgeleitet. Jede Metrik wird zu jeder Cache-Knotenebene berechnet.

Eine vollständige Dokumentation des info Befehls finden Sie unter <http://valkey.io/commands/info>.

Weitere Informationen finden Sie auch unter:

- [Metriken auf Host-Ebene](#)

Metrik	Beschreibung	Einheit
ActiveDefragHits	Die Anzahl der Werteneuzuweisungen pro Minute, die der aktive Defragmentierungsprozess durchführt. Dies wird aus der active_defrag_hits Statistik von abgeleitet. <a href="#">INFO</a>	Zahl
AuthenticationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche, sich mithilfe des Befehls bei Valkey oder Redis OSS zu authentifizieren. AUTH Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem Befehl. <a href="#">ACLLOG</a> Wir empfehlen, hierauf einen Alarm	Anzahl

Metrik	Beschreibung	Einheit
	zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	
	Die Gesamtzahl der Byte, die von Valkey oder Redis OSS für alle Zwecke zugewiesen wurden, einschließlich des Datensatzes, der Puffer usw.	Bytes
BytesUsedForCache	Dimension: Tier=Memory für Valkey- oder OSS Redis-Cluster mit <a href="#">Daten-Tiering ElastiCache</a> : Die Gesamtzahl der für den Cache verwendeten Byte pro Speicher. Dies ist der Wert der <code>used_memory</code> Statistik bei. <a href="#">INFO</a>	Bytes
	Dimension: Tier=SSD für Valkey- oder OSS Redis-Cluster mit <a href="#">Daten-Tiering ElastiCache</a> : Die Gesamtzahl der für den Cache verwendeten Byte von. SSD	Bytes
BytesReadFromDisk	Gesamtzahl der von der Festplatte pro Minute gelesenen Bytes. Wird nur für Cluster mit <a href="#">Daten-Tiering ElastiCache</a> unterstützt.	Bytes
BytesWrittenToDisk	Gesamtzahl der Bytes, die pro Minute auf den Datenträger geschrieben werden. Wird nur für Cluster mit <a href="#">Daten-Tiering ElastiCache</a> unterstützt.	Bytes
CacheHits	Die Anzahl der erfolgreichen schreibgeschützten Schlüsselsuchereignisse im Hauptverzeichnis. Dies wird aus der <code>keyspace_hits</code> Statistik von abgeleitet. <a href="#">INFO</a>	Anzahl

Metrik	Beschreibung	Einheit
CacheMisses	Die Anzahl der nicht erfolgreichen schreibgeschützten Schlüsselsuchereignisse im Hauptverzeichnis. Dies wird aus der <code>keyspace_misses</code> Statistik bei abgeleitet. <a href="#">INFO</a>	Anzahl
CommandAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, Befehle auszuführen, für deren Aufruf sie keine Berechtigung haben. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem <a href="#">ACLLOG</a> Befehl. Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl
CacheHitRate	Zeigt die Nutzungseffizienz der Valkey- oder OSS Redis-Instanz an. Wenn das Cache-Verhältnis unter $\sim 0,8$ liegt, bedeutet dies, dass eine erhebliche Anzahl von Schlüsseln bereinigt wurde, abgelaufen oder nicht vorhanden ist. Dies wird mit <code>cache_hits</code> - und <code>cache_misses</code> -Statistiken auf folgende Weise berechnet: $\text{cache\_hits} / (\text{cache\_hits} + \text{cache\_misses})$ .	Prozent
ChannelAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, auf Kanäle zuzugreifen, für die sie keine Zugriffsberechtigung haben. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem <a href="#">ACLLOG</a> Befehl. Wir empfehlen, für diese Metrik einen Alarm festzulegen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl

Metrik	Beschreibung	Einheit
<code>CurrConnections</code>	Die Anzahl der Client-Verbindungen, ausgenommen Verbindungen von Read Replicas. ElastiCache verwendet jeweils zwei bis vier der Verbindungen, um den Cluster zu überwachen. Dies ergibt sich aus der <code>connected_clients</code> Statistik unter <a href="#">INFO</a> .	Anzahl
<code>CurrItems</code>	Anzahl der Elemente im Cache. Dies wird aus der <code>keyspace</code> Statistik abgeleitet, bei der alle Schlüssel im gesamten Schlüsselraum summiert werden.	Anzahl
	Dimension: <code>Tier=Memory</code> für Cluster mit <a href="#">Daten-Tiering ElastiCache</a> . Anzahl der Elemente im Speicher.	Anzahl
	Dimension: <code>Tier=SSD</code> (Solid-State-Laufwerke) für Cluster mit <a href="#">Daten-Tiering ElastiCache</a> . Die Anzahl der Elemente in SSD.	Anzahl
<code>CurrVolatileItems</code>	Gesamtzahl der Schlüssel in allen Datenbanken mit festgelegtem TTL. Dies wird aus der <code>expires</code> Statistik abgeleitet, bei der alle Schlüssel mit einem TTL-Satz im gesamten Schlüsselraum summiert werden.	Anzahl

Metrik	Beschreibung	Einheit
DatabaseCapacityUsagePercentage	<p>Prozentsatz der gesamten Datenkapazität für den Cluster, die genutzt wird.</p> <p>Bei Data-Tiered-Instances wird die Metrik anhand der Kriterien(<code>used_memory - mem_not_counted_for_evict + SSD used</code>) / (<code>maxmemory + SSD total capacity</code>) , Herkunft und Herkunft <code>used_memory</code> berechnet. <code>maxmemory</code> <a href="#">INFO</a></p> <p>In allen anderen Fällen wird die Metrik anhand von <code>used_memory/maxmemory</code> berechnet.</p>	Prozent
DatabaseCapacityUsageCountedForEvictPercentage	<p>Prozentsatz der gesamten Datenkapazität für den Cluster, die verwendet wird, ohne den für Overhead und verwendeten Speicher COB. Diese Metrik wird wie folgt berechnet:</p> $\frac{\text{used\_memory} - \text{mem\_not\_counted\_for\_evict}}{\text{maxmemory}}$ <p>Bei Daten-Tiering-Instances wird die Metrik wie folgt berechnet:</p> $\frac{(\text{used\_memory} + \text{SSD used})}{(\text{maxmemory} + \text{SSD total capacity})}$ <p>woher <code>used_memory</code> und <code>maxmemory</code> stammen <a href="#">INFO</a></p>	Prozent
DatabaseMemoryUsagePercentage	<p>Prozentsatz des Speichers für den Cluster, der genutzt wird. Dies wird mit Hilfe <code>used_memory/maxmemory</code> von berechnet <a href="#">INFO</a>.</p>	Prozent

Metrik	Beschreibung	Einheit
DatabaseMemoryUsageCountedForEvictPercentage	Prozentsatz des Speichers für den Cluster, der verwendet wird, mit Ausnahme des Speichers, der für Overhead und verwendet wird COB. Dieser Wert wird mithilfe <code>used_memory - mem_not_counted_for_evict / maxmemory</code> von berechnet <a href="#">INFO</a> .	Prozent
DB0AverageTTL	Macht <code>avg_ttl</code> of DBO aus der <code>keyspace</code> Befehlsstatistik verfügbar. <a href="#">INFO</a> Bei Replikaten laufen die Schlüssel nicht ab, die Replikate warten, bis die Schlüssel auf den Primärknoten ablaufen. Wenn ein primärer Knoten einen Schlüssel abläuft (oder ihn aus einem anderen Grund entferntLRU), synthetisiert er einen DEL Befehl, der an alle Replikate übertragen wird. Daher TTL ist <code>db0Average</code> für Replikatknoten 0, da sie Schlüssel nicht ablaufen lassen und somit auch nicht nachverfolgen. TTL	Millisekunden

Metrik	Beschreibung	Einheit
EngineCPUUtilization	<p>Ermöglicht die CPU Nutzung des Valkey- oder Redis-Engine-Threads. OSS Da Valkey und Redis OSS Single-Threading verwenden , können Sie diese Metrik verwenden, um die Auslastung des Prozesses selbst zu analysieren. Die EngineCPUUtilization Metrik bietet eine genauere Darstellung des Prozesses. Sie können sie in Verbindung mit der CPUUtilization Metrik verwenden. CPUUtilization zeigt die CPU Auslastung der Serverinstanz als Ganzes, einschließlich anderer Betriebssystem- und Verwaltungsprozesse. Verwenden Sie bei größeren Knotentypen mit vier vCPUs oder mehr Knoten die EngineCPUUtilization Metrik, um Schwellenwerte für die Skalierung zu überwachen und festzulegen.</p>	Prozent

 **Note**

Auf einem ElastiCache Host überwachen Hintergrundprozesse den Host, um eine verwaltete Datenbankumgebung zu gewährleisten. Diese Hintergrundprozesse können einen erheblichen Teil der CPU Arbeitslast in Anspruch nehmen. Bei größeren Hosts mit mehr als zwei ist dies nicht signifikant vCPUs. Es kann jedoch kleinere Hosts mit 2 vCPUs oder weniger betreffen. Wenn Sie nur die EngineCPUUtilization Metrik überwachen, sind Ihnen Situationen nicht bewusst, in denen der Host sowohl aufgrund der hohen Auslastung durch Valkey oder Redis

Metrik	Beschreibung	Einheit
	OSS als auch aufgrund der hohen CPU Auslastung durch die CPU Hintergrundüberwachungsprozesse überlastet ist. Daher empfehlen wir, die CPUUtilization Metrik für Hosts mit zwei oder weniger zu überwachen. vCPUs	
Evictions	Anzahl der Schlüssel, die infolge des maxmemory -Grenzwertes bereinigt worden sind. Dies wird aus der evicted_keys Statistik unter <a href="#">INFO</a> abgeleitet.	Anzahl
GlobalDatastoreReplicationLag	Dies ist die Verzögerung zwischen dem Primärknoten der sekundären Region und dem Primärknoten der primären Region. Bei aktiviertem Clustermodus Valkey oder Redis OSS gibt die Verzögerung die maximale Verzögerung zwischen den Shards an.	Sekunden
IamAuthenticationExpirations	Die Gesamtzahl der abgelaufenen, IAM authentifizierten Valkey- oder Redis-Verbindungen. OSS Weitere Informationen über <a href="#">Authentifizierung mit IAM</a> finden Sie im Benutzerhandbuch.	Anzahl
IamAuthenticationThrottling	Die Gesamtzahl der gedrosselten, authentifizierten Valkey- oder Redis IAM - oder Anfragen. OSS AUTH HELLO Weitere Informationen über <a href="#">Authentifizierung mit IAM</a> finden Sie im Benutzerhandbuch.	Anzahl
IsMaster	Gibt an, ob der Knoten der Primärknoten des aktuellen Shard/Clusters ist. Die Metrik kann entweder 0 (nicht primär) oder 1 (primär) sein.	Anzahl

Metrik	Beschreibung	Einheit
KeyAuthorizationFailures	Die Gesamtzahl der fehlgeschlagenen Versuche von Benutzern, auf Schlüssel zuzugreifen, für die sie keine Zugriffsberechtigung haben. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem Befehl. <a href="#">ACLLOG</a> Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen.	Anzahl
KeysTracked	Die Anzahl der Schlüssel, die von Valkey oder Redis OSS Key Tracking verfolgt werden, als Prozentsatz von. <code>tracking-table-max-keys</code> Die Schlüsselverfolgung wird verwendet, um das clientseitige Caching zu unterstützen und Clients zu benachrichtigen, wenn Schlüssel geändert werden.	Anzahl
MemoryFragmentationRatio	Gibt die Effizienz bei der Speicherzuweisung der Valkey- oder Redis-Engine an. OSS Bestimmte Schwellenwerte weisen auf unterschiedliche Verhaltensweisen hin. Der empfohlene Wert ist eine Fragmentierung über 1,0. Dies wird aus dem Wert von <code>mem_fragmentation_ratio</code> berechnet. <a href="#">INFO</a>	Zahl

Metrik	Beschreibung	Einheit
NewConnections	<p>Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind. Dies wird aus der <code>total_connections_received</code> Statistik von abgeleitet. <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>Wenn Sie ElastiCache (RedisOSS) Version 5 oder niedriger verwenden, werden zwischen zwei und vier der von dieser Metrik gemeldeten Verbindungen ElastiCache zur Überwachung des Clusters verwendet. Wenn Sie jedoch ElastiCache (RedisOSS) Version 6 oder höher verwenden, sind die Verbindungen, die ElastiCache zur Überwachung des Clusters verwendet werden, nicht in dieser Metrik enthalten.</p> </div>	Anzahl
NumItemsReadFromDisk	Die Gesamtzahl der pro Minute von der Festplatte abgerufenen Elemente. Wird nur für Cluster mit <a href="#">Daten-Tiering ElastiCache</a> unterstützt.	Anzahl
NumItemsWrittenToDisk	Die Gesamtzahl der pro Minute auf die Festplatte geschriebenen Elemente. Wird nur für Cluster mit <a href="#">Daten-Tiering ElastiCache</a> unterstützt.	Anzahl

Metrik	Beschreibung	Einheit
MasterLinkHealthStatus	Dieser Status kann zwei Werte annehmen: 0 oder 1. Der Wert 0 gibt an, dass die Daten im ElastiCache Primärknoten nicht synchron sind, wenn Valkey oder OSS Redis aktiviert sind. EC2 Der Wert 1 bedeutet, dass die Daten synchronisiert sind. Verwenden Sie den Vorgang, um die Migration abzuschließen. <a href="#">CompleteMigrationAPI</a>	Boolesch
Reclaimed	Gesamtanzahl der Schlüsselablaufereignisse Dies wird aus der <code>expired_keys</code> Statistik von abgeleitet. <a href="#">INFO</a>	Anzahl
ReplicationBytes	Für Knoten in einer replizierten Konfiguration gibt <code>ReplicationBytes</code> die Anzahl der Bytes an, die der Primärknoten an alle seine Replikationen sendet. Diese Metrik gibt die Schreiblast auf der Replikationsgruppe an. Dies wird aus der <code>master_repl_offset</code> Statistik bei abgeleitet. <a href="#">INFO</a>	Bytes
ReplicationLag	Diese Metrik ist nur für einen als Read Replica laufenden Knoten verfügbar. Sie stellt die Zeitverzögerung in Sekunden dar, mit der die Replica die vom primären Knoten kommenden Änderungen anwendet. Für Valkey 7.2 und höher sowie Redis OSS Engine-Version 5.0.6 und höher kann die Verzögerung in Millisekunden gemessen werden.	Sekunden

Metrik	Beschreibung	Einheit
SaveInProgress	Diese binäre Metrik gibt 1 zurück, wenn ein Hintergrundspeichervorgang (vergabelt oder unvergabelt) ausgeführt wird, andernfalls beträgt der Wert 0. Ein Hintergrundspeichervorgang wird normalerweise während Snapshots und Synchronisierungen verwendet. Diese Vorgänge können die Leistung beeinträchtigen. Mithilfe der SaveInProgress - Metrik können Sie diagnostizieren, ob die Leistungsbeeinträchtigung von einem Hintergrundspeichervorgang verursacht wurde. Dies wird aus der Statistik von <code>redis_bgsave_in_progress</code> abgeleitet. <a href="#">INFO</a>	Boolesch

Metrik	Beschreibung	Einheit
TrafficManagementActive	<p>Zeigt an, ob ElastiCache (RedisOSS) den Datenverkehr aktiv verwaltet, indem der Datenverkehr, der eingehenden Befehlen zugewiesen wird, überwacht oder repliziert wird. Der Datenverkehr wird verwaltet, wenn mehr Befehle an den Knoten gesendet werden, als von Valkey oder Redis verarbeitet werden können, OSS und dient dazu, die Stabilität und den optimalen Betrieb der Engine aufrecht zu erhalten. Datenpunkte von 1 können darauf hinweisen, dass der Knoten für die bereitgestellte Workload unterdimensioniert ist.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>Note</b></p> <p>Wenn diese Metrik aktiv bleibt, evaluieren Sie den Cluster, um zu entscheiden, ob eine Hoch- oder Aufskalierung erforderlich ist. Zu den verwandten Metriken gehören <code>NetworkBandwidthOutAllowanceExceeded</code> und <code>EngineCPUUtilization</code>.</p> </div>	Boolesch

## E-Verfügbarkeit ngineCPUUtilization

AWS Die unten aufgeführten Regionen sind für alle unterstützten Knotentypen verfügbar.

Region	Name der Region
us-east-2	US East (Ohio)
us-east-1	USA Ost (Nord-Virginia)

Region	Name der Region
us-west-1	USA West (Nordkalifornien)
us-west-2	USA West (Oregon)
ap-northeast-1	Asien-Pazifik (Tokio)
ap-northeast-2	Asien-Pazifik (Seoul)
ap-northeast-3	Asien-Pazifik (Osaka)
ap-east-1	Asien-Pazifik (Hongkong)
ap-south-1	Asien-Pazifik (Mumbai)
ap-southeast-1	Asien-Pazifik (Singapur)
ap-southeast-2	Asien-Pazifik (Sydney)
ap-southeast-3	Asien-Pazifik (Jakarta)
ca-central-1	Kanada (Zentral)
cn-north-1	China (Peking)
cn-northwest-2	China (Ningxia)
me-south-1	Naher Osten (Bahrain)
eu-central-1	Europe (Frankfurt)
eu-west-1	Europa (Irland)
eu-west-2	Europe (London)
eu-west-3	EU (Paris)
eu-south-1	Europa (Milan)
af-south-1	Afrika (Kapstadt)

Region	Name der Region
eu-north-1	Europa (Stockholm)
sa-east-1	Südamerika (São Paulo)
us-gov-west-1	AWS GovCloud (US-West)
us-gov-east-1	AWS GovCloud (US-Ost)

Im Folgenden finden Sie Zusammenfassungen bestimmter Befehle, die von `info commandstats` abgeleitet sind. Der Abschnitt `commandstats` enthält Statistiken, die auf dem Befehlstyp basieren, einschließlich der Anzahl der Aufrufe, der CPU Gesamtzeit, die für diese Befehle benötigt wird, und des durchschnittlichen Zeitaufwands CPU pro Befehlsausführung. Für jeden Befehlstyp wird die folgende Zeile hinzugefügt: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

Die unten aufgeführten Latenzmetriken werden anhand der `Commandstats`-Statistik von berechnet. [INFO](#) Diese werden auf folgende Weise berechnet:  $\text{delta}(\text{usec})/\text{delta}(\text{calls})$ . `delta` wird als Differenz innerhalb einer Minute berechnet. Die Latenz ist definiert als die CPU Zeit, die für die Verarbeitung des ElastiCache Befehls benötigt wird. Beachten Sie, dass bei Clustern, die Daten-Tiering verwenden, die Zeit, die zum Abrufen von Elementen benötigt SSD wird, nicht in diesen Messungen enthalten ist.

Eine vollständige Liste der verfügbaren Befehle finden Sie in der Valkey-Dokumentation unter [Befehle](#).

Metrik	Beschreibung	Einheit
<code>ClusterBasedCmds</code>	Die Gesamtanzahl der Cluster-basierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die auf einen Cluster einwirken ( <code>cluster slotcluster info</code> , usw.).	Anzahl
<code>ClusterBasedCmdsLatency</code>	Latenz von Cluster-basierten Befehlen.	Mikrosekunden

Metrik	Beschreibung	Einheit
EvalBasedCmds	Die Gesamtzahl der Befehle für EVAL-basierte Befehle. Dies wird aus der commandstats Statistik durch Summieren von, abgeleitet. eval evalsha	Anzahl
EvalBasedCmdsLatency	Latenz von eval-basierten Befehlen.	Mikrosekunden
GeoSpatialBasedCmds	Die Gesamtzahl der Befehle für raumbezogene Befehle. Dies wird aus der commandstats Statistik abgeleitet. Es wird abgeleitet, indem alle Befehle des Geo-Typs summiert werden: geoadd, geodist, geohash, geopos, georadius und georadiusbymember.	Anzahl
GeoSpatialBasedCmdsLatency	Latenz von raumbezogenen Befehlen.	Mikrosekunden
GetTypeCmds	Gesamtanzahl der auf read-only basierenden Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle read-only Typbefehle (get,, hget scardlrange, usw.) summiert werden.	Anzahl
GetTypeCmdsLatency	Latenz von Lesebefehlen.	Mikrosekunden
HashBasedCmds	Gesamtanzahl der Hash-basierten Befehle. Dies wird aus der commandstats Statistik abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Hashes (hget,, hkeys hvalshdel, usw.) einwirken.	Anzahl
HashBasedCmdsLatency	Latenz von Hash-basierten Befehlen.	Mikrosekunden

Metrik	Beschreibung	Einheit
HyperLogLogBasedCmds	Gesamtanzahl der auf HyperLogLog basierenden Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehlstypen ( <code>pfadd</code> , <code>pfcountpfmerge</code> , usw.) summiert werden. <code>pf</code>	Anzahl
HyperLogLogBasedCmdsLatency	Latenz von HyperLogLog basierten Befehlen.	Mikrosekunden
JsonBasedCmds	Die Gesamtzahl der JSON Befehle, einschließlich Lese- und Schreibbefehlen. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle JSON Befehle summiert werden, die sich auf JSON Tasten auswirken.	Anzahl
JsonBasedCmdsLatency	Latenz aller JSON Befehle, einschließlich Lese- und Schreibbefehlen.	Mikrosekunden
JsonBasedGetCmds	Die Gesamtzahl der Befehle, die JSON nur lesbar sind. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle JSON Lesebefehle summiert werden, die sich auf Tasten auswirken. <code>JSON</code>	Anzahl
JsonBasedGetCmdsLatency	Latenz von Befehlen, die nur JSON lesbar sind.	Mikrosekunden
JsonBasedSetCmds	Die Gesamtzahl der JSON Schreibbefehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle JSON Schreibbefehle summiert werden, die auf JSON Tasten wirken.	Anzahl
JsonBasedSetCmdsLatency	Latenz der JSON Schreibbefehle.	Mikrosekunden

Metrik	Beschreibung	Einheit
KeyBasedCmds	Gesamtanzahl der schlüsselbasierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die auf einen oder mehrere Schlüssel in mehreren Datenstrukturen ( <code>delexpire</code> , <code>rename</code> , usw.) einwirken.	Anzahl
KeyBasedCmdsLatency	Latenz von schlüsselbasierten Befehlen.	Mikrosekunden
ListBasedCmds	Gesamtanzahl der listenbasierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf eine oder mehrere Listen auswirken ( <code>lindex</code> , <code>lrange</code> <code>lpushltrim</code> , usw.).	Anzahl
ListBasedCmdsLatency	Latenz von listenbasierten Befehlen.	Mikrosekunden
NonKeyTypeCmds	Gesamtanzahl der nicht schlüsselbasierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich nicht auf eine Taste auswirken, z. B., <code>acl</code> <code>dbsize</code> <code>info</code>	Anzahl
NonKeyTypeCmdsLatency	Latenz der Befehle. non-key-based	Mikrosekunden
PubSubBasedCmds	Die Gesamtzahl der Befehle für die Pub/Sub-Funktionalität. Dies wird aus den <code>commandstats</code> Statistiken abgeleitet, indem alle Befehle zusammengefasst werden, die für die Pub/Sub-Funktionalität verwendet werden: <code>psubscribe</code> , <code>publish</code> , <code>pubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> , <code>sunsubscribe</code> <code>publish</code> , <code>subscribe</code> und <code>unsubscribe</code>	Anzahl

Metrik	Beschreibung	Einheit
PubSubBasedCmdsLatency	Latenz von pub/sub-basierten Befehlen.	Mikrosekunden
SetBasedCmds	Gesamtanzahl der Set-basierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf einen oder mehrere Sätze auswirken ( <code>scard</code> , <code>sdiff</code> <code>saddunion</code> , usw.).	Anzahl
SetBasedCmdsLatency	Latenz von Set-basierten Befehlen.	Mikrosekunden
SetTypeCmds	Gesamtanzahl der auf write basierenden Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle mutative Befehlstypen summiert werden, die mit Daten arbeiten ( <code>set</code> , <code>hset</code> , <code>saddlpop</code> , usw.)	Anzahl
SetTypeCmdsLatency	Latenz von Schreibbefehlen.	Mikrosekunden
SortedSetBasedCmds	Gesamtanzahl der Sorted Set-basierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf eine oder mehrere sortierte Sätze ( <code>zcount</code> , <code>zrange</code> <code>zrankzadd</code> , usw.) auswirken.	Anzahl
SortedSetBasedCmdsLatency	Latenz von Sortierungs-basierten Befehlen.	Mikrosekunden

Metrik	Beschreibung	Einheit
StringBasedCmds	Gesamtanzahl der Zeichenfolge-basierten Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf eine oder mehrere Zeichenketten ( <code>strlen</code> , <code>setexsetrange</code> , usw.) auswirken.	Anzahl
StringBasedCmdsLatency	Latenz von Zeichenfolgen-basierten Befehlen.	Mikrosekunden
StreamBasedCmds	Die Gesamtanzahl Stream-basierter Befehle. Dies wird aus der <code>commandstats</code> Statistik abgeleitet, indem alle Befehle summiert werden, die sich auf einen oder mehrere Stream-Datentypen ( <code>xrange</code> , <code>xlen</code> , <code>xaddxdel</code> , usw.) auswirken.	Anzahl
StreamBasedCmdsLatency	Latenz von Stream-basierten Befehlen.	Mikrosekunden

## Metriken für Memcached

Der `AWS/ElastiCache`-Namespace enthält die folgenden Memcache-Metriken.

Der ElastiCache Namespace `AWS/`enthält die folgenden Metriken, die vom Befehl `Memcached stats` abgeleitet sind. Jede Metrik wird zu jeder Cache-Knotenebene berechnet.

Informationen finden Sie auch unter:

- [Metriken auf Host-Ebene](#)

Metrik	Beschreibung	Einheit
BytesReadIntoMemcached	Anzahl der Byte, die vom Cache-Knoten aus dem Netzwerk gelesen wurden.	Bytes

Metrik	Beschreibung	Einheit
BytesUsedForCacheItems	Anzahl der Byte, die zum Speichern von Cache-Elementen verwendet werden.	Bytes
BytesWrittenOutFromMemcached	Anzahl der Byte, die vom Cache-Knoten in das Netzwerk geschrieben wurden.	Bytes
CasBadval	Die Anzahl der Anfragen CAS (Check and Set), die der Cache erhalten hat und bei denen der Cas-Wert nicht mit dem gespeicherten Cas-Wert übereinstimmt.	Anzahl
CasHits	Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel gefunden wurde und die CAS-Werte übereinstimmen.	Anzahl
CasMisses	Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
CmdFlush	Anzahl der flush-Befehle, die der Cache erhalten hat.	Anzahl
CmdGet	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
CmdSet	Die Anzahl der Set-Befehle, die der Cache empfangen hat.	Anzahl

Metrik	Beschreibung	Einheit
<code>CurrConnections</code>	<p>Die Anzahl der zu einem bestimmten Zeitpunkt mit dem Cache verbundenen Verbindungen. ElastiCache verwendet zwei bis drei der Verbindungen, um den Cluster zu überwachen.</p> <p>Zusätzlich zu den oben genannten erstellt memcached eine Anzahl von internen Verbindungen, die dem Doppelten der für den Knotentyp verwendeten Threads entsprechen. Die Thread-Anzahl für die verschiedenen Knotentypen ist in <code>Nodetype Specific Parameters</code> der entsprechenden Parametergruppe zu sehen.</p> <p>Die Gesamtzahl der Verbindungen ist die Summe der Clientverbindungen, der zu überwachenden Verbindungen und der oben genannten internen Verbindungen.</p>	Anzahl
<code>CurrItems</code>	Die Anzahl der zu einem bestimmten Zeitpunkt mit dem Cache verbundenen Verbindungen.	Anzahl
<code>DecrHits</code>	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
<code>DecrMisses</code>	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
<code>DeleteHits</code>	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl

Metrik	Beschreibung	Einheit
DeleteMisses	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
Evictions	Die Anzahl nicht abgelaufener Elemente, die vom Cache bereinigt wurden, um Platz für neue Schreibvorgänge zu schaffen.	Anzahl
GetHits	Die Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
GetMisses	Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrHits	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrMisses	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
Reclaimed	Die Anzahl abgelaufener Elemente, die vom Cache bereinigt wurden, um Platz für neue Schreibvorgänge zu schaffen.	Anzahl

Für Memcached 1.4.14 stehen die folgenden zusätzlichen Metriken zur Verfügung.

Metrik	Beschreibung	Einheit
BytesUsedForHash	Die Anzahl der Byte, die derzeit von Hash-Tabellen verwendet werden.	Bytes
CmdConfigGet	Die kumulative Anzahl an config get-Anforderungen.	Anzahl
CmdConfigSet	Die kumulative Anzahl an config set-Anforderungen.	Anzahl
CmdTouch	Die kumulative Anzahl an touch-Anforderungen.	Anzahl
CurrConfig	Die aktuelle Anzahl gespeicherter Konfigurationen.	Anzahl
EvictedUnfetched	Die Anzahl der gültigen Elemente, die aus dem Cache (LRU), der am wenigsten benutzt wurde, entfernt wurden und die nach dem Setzen nicht verändert wurden.	Anzahl
ExpiredUnfetched	Die Anzahl der abgelaufenen Elemente, die aus dem zurückgeholt wurden und LRU die nach dem Setzen nicht verändert wurden.	Anzahl
SlabsMoved	Die Gesamtanzahl der Slab Pages, die verschoben worden sind.	Anzahl
TouchHits	Die Anzahl der Schlüssel, auf die zugegriffen wurde und die mit einer neuen Ablaufzeit versehen wurden.	Anzahl
TouchMisses	Die Anzahl der Elemente, auf die zwar zugegriffen wurde, die aber nicht gefunden werden konnten.	Anzahl

Der ElastiCache Namespace AWS/enthält die folgenden berechneten Metriken auf Cache-Ebene.

Metrik	Beschreibung	Einheit
NewConnections	Die Anzahl der neuen Verbindungen, die der Cache erhalten hat. Dies wird von der memcached-Statistik total_connections abgeleitet, indem die Änderung in total_connections über einen bestimmten Zeitraum aufgezeichnet wird. Dieser Wert wird immer mindestens 1 sein, da eine Verbindung für a reserviert ist. ElastiCache	Anzahl
NewItems	Die Anzahl der neuen Elemente, die im Cache gespeichert wurden. Dies wird von der memcached-Statistik total_items abgeleitet, indem die Änderung in total_items über einen bestimmten Zeitraum aufgezeichnet wird.	Anzahl
UnusedMemory	<p>Die Größe des nicht von Daten belegten Speichers. Dies wird von den Memcached-Statistiken limit_maxbytes und bytes abgeleitet, indem bytes von limit_maxbytes subtrahiert werden.</p> <p>Da der Memcached-Overhead zusätzlich zu dem für Daten verwendeten Speicher belegt, UnusedMemory sollte dieser Wert nicht als die Menge an Speicher angesehen werden, die für zusätzliche Daten zur Verfügung steht. Möglicherweise treten Bereinigungen auch dann auf, wenn noch etwas ungenutzter Speicherplatz vorhanden ist.</p> <p>Detailliertere Informationen erhalten Sie unter <a href="#">Memcached item memory usage</a>.</p>	Bytes

## Welche Metriken sollte ich überwachen?

Die folgenden CloudWatch Kennzahlen bieten einen guten Einblick in die ElastiCache Leistung. In den meisten Fällen empfehlen wir, CloudWatch Alarme für diese Kennzahlen einzurichten, damit Sie Korrekturmaßnahmen ergreifen können, bevor Leistungsprobleme auftreten.

Zu überwachende Metriken

- [CPUUtilization](#)
- [EngineCPUUtilization](#)
- [SwapUsage \(Valkey und Redis\) OSS](#)
- [Evictions](#)
- [CurrConnections](#)
- [Speicher \(Valkey und Redis\) OSS](#)
- [Netzwerk](#)
- [Latency](#)
- [Replikation](#)
- [Verkehrsmanagement \(Valkey und Redis\) OSS](#)

### CPUUtilization

Diese Metrik auf Hostebene wird in Prozent angegeben. Weitere Informationen finden Sie unter [Metriken auf Host-Ebene](#).

Valkey und Redis OSS

Verwenden Sie bei kleineren Knotentypen mit 2 vCPUs oder weniger die `CPUUtilization` Metrik, um Ihre Arbeitslast zu überwachen.

Im Allgemeinen empfehlen wir, dass Sie Ihren Schwellenwert auf 90% Ihres verfügbaren Schwellenwerts festlegen CPU. Da Valkey und Redis OSS beide Single-Threading verwenden, sollte der tatsächliche Schwellenwert als Bruchteil der Gesamtkapazität des Knotens berechnet werden. Angenommen, Sie verwenden einen Knotentyp mit zwei Kernen. In diesem Fall `CPUUtilization` wäre der Schwellenwert für  $90/2$  oder 45%.

Sie müssen eigene Grenzwerte basierend auf der Anzahl der Kerne im verwendeten Cache-Knoten festlegen. Wenn dieser Grenzwert überschritten wird und der Workload hauptsächlich von

Leseanfragen stammt, erweitern Sie den Cache-Cluster, indem Sie Read Replicas hinzufügen. Wenn der Workload hauptsächlich aus Schreibanfragen stammt, empfehlen wir Ihnen abhängig von Ihrer Cluster-Konfiguration:

- Valkey- oder Redis-Cluster OSS (Cluster-Modus deaktiviert): Skalieren Sie, indem Sie einen größeren Cache-Instance-Typ verwenden.
- Valkey- oder Redis-Cluster OSS (Clustermodus aktiviert): Fügen Sie weitere Shards hinzu, um die Schreiblast auf mehr Primärknoten zu verteilen.

### Tip

Anstatt die Metrik auf Host-Ebene zu verwenden `CPUUtilization`, können Valkey- und OSS Redis-Benutzer möglicherweise die Metrik verwenden `EngineCPUUtilization`, die den Prozentsatz der Nutzung auf dem Valkey- oder Redis-Engine-Kern angibt. OSS [Um zu sehen, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere Informationen finden Sie unter Metriken für Valkey und Redis. OSS](#)

Für größere Knotentypen mit 4 vCPUs oder mehr können Sie die `EngineCPUUtilization` Metrik verwenden, die den Prozentsatz der Nutzung auf dem Valkey- oder OSS Redis-Engine-Kern angibt. Um zu sehen, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere Informationen finden Sie unter [Metriken für Redis. OSS](#)

## Memcached

Da Memcached mit mehreren Threads arbeitet, darf diese Metrik bis zu 90 % erreichen. Wenn Sie diesen Schwellenwert überschreiten, skalieren Sie Ihren Cache-Cluster, indem Sie einen größeren Cache-Knotentyp verwenden, oder skalieren Sie ihn, indem Sie weitere Cache-Knoten hinzufügen.

## EngineCPUUtilization

Für größere Knotentypen mit 4 vCPUs oder mehr können Sie die `EngineCPUUtilization` Metrik verwenden, die den Prozentsatz der Nutzung auf dem OSS Redis-Engine-Kern angibt. Um zu sehen, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere Informationen finden Sie unter [Metriken für Valkey und Redis. OSS](#)

Weitere Informationen finden Sie im CPUsAbschnitt [Bewährte Methoden zur Überwachung mit Amazon ElastiCache \(RedisOSS\) mithilfe von Amazon CloudWatch](#).

## SwapUsage (Valkey und Redis) OSS

Diese Metrik auf Hostebene wird in Bytes angegeben. Weitere Informationen finden Sie unter [Metriken auf Host-Ebene](#).

Liegt die `FreeableMemory` CloudWatch Metrik nahe 0 (d. h. unter 100 MB) oder ist sie größer als die `SwapUsage FreeableMemory` Metrik, bedeutet dies, dass ein Knoten unter Speicherauslastung steht. Beachten Sie in diesem Fall folgende Themen

- [Stellen Sie sicher, dass Sie über genügend Speicherplatz verfügen, um einen Valkey- oder OSS Redis-Snapshot zu erstellen](#)
- [Verwaltung des reservierten Speichers für Valkey und Redis OSS](#)

## Evictions

Dies ist eine Metrik für die Cache-Engine. Wir empfehlen Ihnen, einen eigenen Grenzwert für diese Metrik basierend auf den Anforderungen Ihrer Anwendung zu bestimmen.

Wenn Sie Memcached verwenden und den von Ihnen gewählten Schwellenwert überschreiten, skalieren Sie Ihren Cluster, indem Sie einen größeren Knotentyp verwenden, oder skalieren Sie, indem Sie weitere Knoten hinzufügen.

## CurrConnections

Dies ist eine Metrik für die Cache-Engine. Wir empfehlen Ihnen, einen eigenen Grenzwert für diese Metrik basierend auf den Anforderungen Ihrer Anwendung zu bestimmen.

Eine zunehmende Anzahl von `CurrConnections` kann auf ein Problem mit Ihrer Anwendung hinweisen. Um dieses Problem zu beheben, müssen Sie das Verhalten der Anwendung untersuchen.

Weitere Informationen finden Sie im Abschnitt Verbindungen unter [Bewährte Methoden zur Überwachung von Amazon ElastiCache \(RedisOSS\) mithilfe von Amazon CloudWatch](#).

## Speicher (Valkey und Redis) OSS

Speicher ist ein Kernaspekt von Valkey und Redis. OSS Es ist notwendig, die Speicherauslastung Ihres Clusters zu verstehen, um Datenverluste zu vermeiden und das zukünftige Wachstum Ihres Datasets berücksichtigen zu können. Statistiken über die Speicherauslastung eines Knotens sind im Speicherbereich des `INFO` Befehls verfügbar.

Weitere Informationen finden Sie im Abschnitt Speicher unter [Bewährte Methoden zur Überwachung mit Amazon ElastiCache \(RedisOSS\) mithilfe von Amazon CloudWatch](#).

## Netzwerk

Einer der entscheidenden Faktoren für die Kapazität der Netzwerkbandbreite Ihres Clusters ist der von Ihnen ausgewählte Knotentyp. Weitere Informationen zur Netzwerkkapazität Ihres Nodes finden Sie unter [ElastiCache Amazon-Preise](#).

Weitere Informationen finden Sie im Abschnitt Netzwerk unter [Bewährte Methoden zur Überwachung mit Amazon ElastiCache \(RedisOSS\) mithilfe von Amazon CloudWatch](#).

## Latency

Sie können die Latenz eines Befehls anhand einer Reihe von CloudWatch Metriken messen, die aggregierte Latenzen pro Datenstruktur angeben. Diese Latenzmetriken werden anhand der `commandstats` Statistik aus dem Valkey-Befehl berechnet. [INFO](#)

Weitere Informationen finden Sie im Abschnitt Latenz unter [Bewährte Methoden zur Überwachung mit Amazon ElastiCache mithilfe von Amazon CloudWatch](#).

## Replikation

Das Datenvolumen, das repliziert wird, ist über die `ReplicationBytes`-Metrik ersichtbar. Obwohl diese Metrik für die Schreiblast der Replikationsgruppe repräsentativ ist, gibt sie keine Einblicke in den Replikationsstatus. Für diesen Zweck können Sie die `ReplicationLag`-Metrik verwenden.

Weitere Informationen finden Sie im Abschnitt Replikation unter [Bewährte Methoden zur Überwachung mit Amazon ElastiCache \(RedisOSS\) mithilfe von Amazon CloudWatch](#).

## Verkehrsmanagement (Valkey und Redis) OSS

ElastiCache (RedisOSS) verwaltet automatisch den Datenverkehr für einen Knoten, wenn mehr eingehende Befehle an den Knoten gesendet werden, als von Valkey oder Redis verarbeitet werden können. OSS Dies geschieht, um den optimalen Betrieb und die Stabilität der Engine aufrechtzuerhalten.

Wenn der Datenverkehr auf einem Knoten aktiv verwaltet wird, gibt die Metrik `TrafficManagementActive` den Datenpunkt 1 aus. Dies weist darauf hin, dass der Knoten für den bereitgestellten Workload möglicherweise unterskaliert ist. Wenn diese Metrik über einen längeren

Zeitraum 1 bleibt, evaluieren Sie den Cluster, um zu entscheiden, ob eine Hoch- oder Aufskalierung erforderlich ist.

Sehen Sie sich die Metrik `TrafficManagementActive` auf der Seite [Metriken](#) an, um mehr zu erfahren.

## Auswählen von Metrikstatistiken und -zeiträumen

CloudWatch Sie können zwar eine beliebige Statistik und einen beliebigen Zeitraum für jede Metrik auswählen, aber nicht alle Kombinationen sind nützlich. Beispielsweise CPUUtilization sind die Statistiken „Durchschnitt“, „Minimum“ und „Maximum“ nützlich, die Summenstatistik jedoch nicht.

Alle ElastiCache Samples werden für einen Zeitraum von 60 Sekunden für jeden einzelnen Cache-Knoten veröffentlicht. Eine Metrik für einen Cache-Knoten enthält für einen 60-Sekunden-Zeitraum immer nur eine Stichprobe.

Weitere Informationen zum Abrufen von Metriken für Cache-Knoten finden Sie unter [Überwachung von CloudWatch Cluster- und Knotenmetriken](#).

## Überwachung von CloudWatch Cluster- und Knotenmetriken

ElastiCache und CloudWatch sind integriert, sodass Sie eine Vielzahl von Metriken sammeln können. Sie können diese Metriken überwachen mit CloudWatch.

### Note

Für die folgenden Beispiele sind die CloudWatch Befehlszeilentools erforderlich. Weitere Informationen zu den Entwicklertools CloudWatch und zum Herunterladen finden Sie auf der [CloudWatch Produktseite](#).

Die folgenden Verfahren zeigen Ihnen, wie Sie Speicherplatzstatistiken für einen Cache-Cluster für die letzte Stunde sammeln können. CloudWatch

### Note

Die Werte für `StartTime` und `EndTime` in diesen Beispielen unten dienen nur zur Veranschaulichung. Sie müssen die entsprechenden Werte für den Start- und Endzeitpunkt Ihrer Cache-Knoten einsetzen.

Informationen zu ElastiCache Grenzwerten finden Sie unter [AWS Service Limits](#) für ElastiCache.

## Überwachung von CloudWatch Cluster- und Knotenmetriken (Konsole)

Um CPU Nutzungsstatistiken für einen Cache-Cluster zu sammeln

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie die Cache-Knoten aus, für die Sie Metriken anzeigen möchten.

 Note

Bei der Auswahl von mehr als 20 Knoten wird die Ansicht von Metriken auf der Konsole deaktiviert.

- a. Klicken Sie auf der Seite Cache-Cluster der AWS Management Console auf den Namen eines oder mehrerer Cache-Cluster.  
  
Die Detailseite zum Cache-Cluster wird angezeigt.
- b. Klicken Sie oben im Fenster auf die Registerkarte Nodes.
- c. Wählen Sie auf der Registerkarte Nodes des Detailfensters die Cache-Knoten aus, für die Sie Metriken anzeigen möchten.

Am unteren Rand des Konsolenfensters wird eine Liste der verfügbaren CloudWatch Metriken angezeigt.

- d. Klicken Sie auf die CPUNutzungsmetrik.

Die CloudWatch Konsole wird geöffnet und zeigt Ihre ausgewählten Metriken an. Sie können die Dropdown-Listenfelder Statistic und Period und die Registerkarte Time Range verwenden, um die angezeigten Metriken zu ändern.

## Überwachen von CloudWatch Cluster- und Knotenmetriken mithilfe der CloudWatch CLI

Um CPU Nutzungsstatistiken für einen Cache-Cluster zu sammeln

- Für Linux, macOS oder Unix:

```
aws cloudwatch get-metric-statistics \
 --namespace AWS/ElastiCache \
 --metric-name CPUUtilization \
 --dimensions='[{"Name":"CacheClusterId","Value":"test"},
{"Name":"CacheNodeId","Value":"0001"}]' \
 --start-time 2015-02-02T00:00:00Z \
 --end-time 2015-02-02T00:00:00Z
```

```
--statistics=Average \
--start-time 2018-07-05T00:00:00 \
--end-time 2018-07-06T00:00:00 \
--period=3600
```

Für 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
```

## Überwachen von CloudWatch Cluster- und Knotenmetriken mit dem CloudWatch API

Um CPU Nutzungsstatistiken für einen Cache-Cluster zu sammeln

- Rufen Sie das CloudWatch API `GetMetricStatistics` mit den folgenden Parametern auf (beachten Sie, dass die Start- und Endzeiten nur als Beispiele angezeigt werden; Sie müssen Ihre eigenen entsprechenden Start- und Endzeiten ersetzen):
  - `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>
```

# Kontingente für ElastiCache

Ihr AWS Konto verfügt über Standardkontingente, die früher als Limits bezeichnet wurden, für jeden AWS Dienst. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Um die Kontingente für anzuzeigen ElastiCache, öffnen Sie die [Konsole Service Quotas](#). Wählen Sie im Navigationsbereich AWS Dienste aus und wählen Sie ElastiCache.

Informationen zur Erhöhung eines Kontingents finden Sie unter [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch zu Service Quotas. Wenn das Kontingent unter Service Quotas noch nicht in verfügbar ist, verwenden Sie das [Formular zur Erhöhung des Service-Limits](#).

Für Ihr AWS Konto gelten die folgenden Kontingente für ElastiCache.

Ressource	Standard
Serverless-Caches pro Region	40
Serverlose Snapshots pro Tag pro Cache, Redis	24
Knoten pro Region	300
Knoten pro Cluster, Memcached	60
Knoten pro Cluster pro Instanztyp, Valkey oder Redis OSS (Clustermodus aktiviert)	90
Knoten pro Shard, Valkey oder Redis OSS (Clustermodus deaktiviert)	6
Parametergruppen pro Region	300
Sicherheitsgruppen pro Region	50
Subnetzgruppen pro Region	300
Subnetze pro Subnetzgruppe	20

Ressource	Standard
Benutzer pro Benutzergruppe, Redis	100
Maximale Anzahl von Benutzern, Redis	1000
Maximale Anzahl von Benutzergruppen, Redis	100

# Referenz

Die Themen in diesem Abschnitt behandeln die Arbeit mit Amazon ElastiCache API und den ElastiCache Abschnitt der AWS CLI. In diesem Abschnitt sind auch allgemeine Fehlermeldungen und Servicemeldungen enthalten.

- [Mit dem ElastiCache API](#)
- [ElastiCache APIReferenz](#)
- [ElastiCache Abschnitt der AWS CLI Referenz](#)
- [ElastiCache Amazon-Fehlermeldungen](#)
- [Benachrichtigungen](#)

## Mit dem ElastiCache API

Dieser Abschnitt enthält aufgabenorientierte Beschreibungen der Verwendung und Implementierung ElastiCache von Vorgängen. Eine vollständige Beschreibung dieser Vorgänge finden Sie in der [ElastiCache APIAmazon-Referenz](#).

Themen

- [Die Abfrage verwenden API](#)
- [Verfügbare Bibliotheken](#)
- [Fehlerbehebung bei Anwendungen](#)

## Die Abfrage verwenden API

### Abfrageparameter

HTTPAbfragebasierte Anfragen sind HTTP Anfragen, die das HTTP Verb GET oder POST und einen Abfrageparameter mit dem Namen verwenden. Action

Jede Query-Anfrage muss einige allgemeine Parameter enthalten, um die Authentifizierung und Auswahl einer Aktion zu bearbeiten.

Einige Operationen verwenden Parameterlisten. Diese Listen werden mit der Notation `param.n` definiert. Werte von `n` sind ganze Zahlen, die bei 1 beginnen.

## Authentifizierung von Abfrageanforderungen

Sie können nur Query-Anfragen senden HTTPS und müssen in jeder Query-Anfrage eine Signatur einschließen. In diesem Abschnitt wird beschrieben, wie Sie die Signatur erstellen. Die in den folgenden Schritten beschriebene Methode wird als Signaturversion 4 bezeichnet.

Die folgenden grundlegenden Schritte dienen der Authentifizierung von Anfragen an AWS. Dies setzt voraus, dass Sie registriert sind AWS und über eine Zugriffsschlüssel-ID und einen geheimen Zugriffsschlüssel verfügen.

### Abfrageauthentifizierungsprozess

1. Der Absender erstellt eine Anfrage an AWS.
2. Der Absender berechnet die Anforderungssignatur, ein Keyed-Hashing for Hash-based Message Authentication Code (HMAC) mit einer Hash-Funktion von SHA -1, wie im nächsten Abschnitt dieses Themas definiert.
3. Der Absender der Anfrage sendet die Anforderungsdaten, die Signatur und die Zugriffsschlüssel-ID (die Schlüssel-ID des verwendeten geheimen Zugriffsschlüssels) an. AWS
4. AWS verwendet die Zugriffsschlüssel-ID, um den geheimen Zugriffsschlüssel nachzuschlagen.
5. AWS generiert eine Signatur aus den Anforderungsdaten und dem geheimen Zugriffsschlüssel unter Verwendung desselben Algorithmus, der zur Berechnung der Signatur in der Anfrage verwendet wurde.
6. Wenn die Signaturen übereinstimmen, wird die Anforderung als authentisch betrachtet. Falls der Vergleich fehlschlägt, wird die Anforderung verworfen, und AWS gibt eine Fehlerantwort zurück.

#### Note

Wenn eine Anforderung einen `Timestamp`-Parameter enthält, läuft die für die Anforderung berechnete Signatur 15 Minuten nach dessen Wert ab.

Wenn eine Anforderung einen `Expires`-Parameter enthält, läuft die Signatur an dem durch den `Expires`-Parameter angegebenen Zeitpunkt ab.

### So wird die Anfragesignatur berechnet

1. Erstellen Sie eine vereinheitlichte Abfragezeichenfolge, die Sie später bei dieser Prozedur benötigen:

- a. Sortiert die Komponenten der Abfragezeichenfolge UTF -8 nach dem Parameternamen mit natürlicher Bytereihenfolge. Die Parameter können aus dem GET URI oder aus dem POST Hauptteil stammen (wenn Content-Type Application/ ist). x-www-form-urlencoded
  - b. URLkodieren Sie den Namen und die Werte des Parameters gemäß den folgenden Regeln:
    - i. URLKodieren Sie keines der nicht reservierten Zeichen, RFC die 3986 definiert. Folgende Zeichen sind nicht reserviert: A – Z, a – z, 0 – 9, Bindestrich ( - ), Unterstrich ( \_ ), Punkt ( . )
    - ii. Versehen Sie alle anderen Zeichen mit Prozentcode (%XY), wobei X und Y für Hexadezimalzeichen, d. h. 0-9 und die Großbuchstaben A-F, steht.
    - iii. Prozent kodieren erweiterte UTF -8 Zeichen in der Form %XY%ZA...
    - iv. Versehen Sie das Leerzeichen mit dem Prozentcode %20 (und nicht + wie in herkömmlichen Codierungsschemata).
  - c. Trennen Sie die kodierten Parameternamen mit dem Gleichheitszeichen (=) (ASCII Zeichen 61) von ihren codierten Werten, auch wenn der Parameterwert leer ist.
  - d. Trennen Sie die Name-Wert-Paare durch ein Und-Zeichen (&) (Code 38). ASCII
2. Erstellen Sie die zu signierende Zeichenfolge gemäß der folgenden Pseudogrammatik (das "\n" steht für einen Zeilenumbruch). ASCII

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

Bei der HTTPRequestURI Komponente handelt es sich um die HTTP absolute Pfadkomponente der URI Abfragezeichenfolge, die bis zu, aber nicht einschließt. Wenn der leer HTTPRequestURI ist, verwenden Sie einen Schrägstrich (/).

3. Berechne ein RFC 2104-konformes Ergebnis HMAC mit der gerade erstellten Zeichenfolge, deinem Secret Access Key als Schlüssel SHA256 und/oder SHA1 als Hash-Algorithmus.

Weitere Informationen finden Sie unter <https://www.ietf.org/rfc/rfc2104.txt>.

4. Konvertieren Sie den resultierenden Wert zu base64.
5. Schließen Sie den Wert als Wert des Signature-Parameters in die Anforderung ein.

Das folgende Beispiel zeigt eine Anforderung (die Zeilenumbrüche wurden aus Gründen der Übersichtlichkeit hinzugefügt).

```
https://elasticache.us-west-2.amazonaws.com/
 ?Action=DescribeCacheClusters
 &CacheClusterIdentifier=myCacheCluster
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &Version=2014-12-01
```

Für die vorherige Abfragezeichenfolge würden Sie die HMAC Signatur anhand der folgenden Zeichenfolge berechnen.

```
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:
```

Das Ergebnis ist die folgende signierte Anforderung.

```
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=2877960fced9040b41b4feaca835fd5cf9264f768e6a0236c9143f915ffa56
```

Ausführliche Informationen über das Signieren und Berechnen der Anforderungssignatur finden Sie im Thema [Signaturprozess mit Signature Version 4](#) und in den entsprechenden Unterthemen.

## Verfügbare Bibliotheken

AWS stellt Software Development Kits (SDKs) für Softwareentwickler bereit, die es vorziehen, Anwendungen sprachspezifisch statt mithilfe von Query APIs zu erstellen. Diese SDKs bieten grundlegende Funktionen (nicht im Lieferumfang enthalten APIs) wie Anforderungsauthentifizierung, Wiederholungsversuche und Fehlerbehandlung, sodass der Einstieg erleichtert wird. SDKs und zusätzliche Ressourcen sind für die folgenden Programmiersprachen verfügbar:

- [Java](#)
- [Windows und .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

Weitere Informationen zu anderen Sprachen finden Sie unter [Beispiel-Code und Bibliotheken](#).

## Fehlerbehebung bei Anwendungen

ElastiCache bietet spezifische und beschreibende Fehler, die Ihnen bei der Behebung von Problemen bei der Interaktion mit dem ElastiCache API helfen sollen.

### Fehler bei Abrufen

In der Regel sollte Ihre Anwendung überprüfen, ob eine Anforderung einen Fehler verursacht hat, bevor Sie Zeit für die Verarbeitung von Ergebnissen aufwenden. Der einfachste Weg, um herauszufinden, ob ein Fehler aufgetreten ist, besteht darin, in der Antwort von nach einem `ERROR` Knoten zu suchen. ElastiCache API

XPath Die Syntax bietet eine einfache Möglichkeit, nach dem Vorhandensein eines `ERROR` Knotens zu suchen und den Fehlercode und die Fehlermeldung abzurufen. Der folgende Codeausschnitt

verwendet Perl und dasXML::XPath -Modul, um festzustellen, ob bei einer Anfrage ein Fehler aufgetreten ist. Wenn ein Fehler aufgetreten ist, gibt der Code den ersten Fehlercode und die erste Fehlermeldung in der Antwort an.

```
use XML::XPath;
my $xp = XML::XPath->new(xml =>$response);
if ($xp->find("//Error"))
{print "There was an error processing your request:\n", " Error code: ",
$xp->findvalue("//Error[1]/Code"), "\n", " ",
$xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

## Tipps zur Problembhebung

Wir empfehlen die folgenden Verfahren zur Diagnose und Lösung von Problemen mit dem ElastiCache API

- Stellen Sie sicher, dass ElastiCache es ordnungsgemäß läuft.

Öffnen Sie dazu einfach ein Browserfenster und senden Sie eine Abfrageanfrage an den ElastiCache Dienst (z. B. <https://elasticache.amazonaws.com>). Ein interner Serverfehler `MissingAuthenticationTokenException` oder 500 bestätigt, dass der Dienst verfügbar ist und auf Anfragen reagiert.

- Überprüfen Sie die Struktur Ihrer Anforderung.

Für jeden ElastiCache Vorgang gibt es in der Referenz eine ElastiCache APIReferenzseite. Prüfen Sie nochmals, dass Sie die Parameter korrekt verwenden. Die Beispielanforderungen oder Benutzerszenarien zeigen Ihnen, ob ähnliche Operationen ausgeführt werden, und vermitteln Ihnen eine Vorstellung von möglichen Fehlern.

- Sehen Sie im Forum nach.

ElastiCache hat ein Diskussionsforum, in dem Sie nach Lösungen für Probleme suchen können, auf die andere unterwegs gestoßen sind. Weitere Informationen zur Anzeige des Forums finden Sie unter

<https://forums.aws.amazon.com/> .

# Einrichtung der ElastiCache Befehlszeilenschnittstelle

In diesem Abschnitt wird beschrieben, welche Voraussetzungen für die Ausführung der Befehlszeilen-Tools erforderlich sind, wo Sie die Befehlszeilen-Tools herunterladen können und wie die Tools und die Umgebung eingerichtet werden. Zudem werden gängige Beispiele für die Nutzung der Tools vorgestellt.

Folgen Sie den Anweisungen in diesem Thema nur, wenn Sie AWS CLI zum Forum wechseln ElastiCache.

## Important

Die ElastiCache Amazon-Befehlszeilenschnittstelle (CLI) unterstützt keine ElastiCache Verbesserungen nach API Version 2014-09-30. Um neuere ElastiCache Funktionen von der Befehlszeile aus zu verwenden, verwenden Sie die [AWS Befehlszeilenschnittstelle](#).

## Themen

- [Voraussetzungen](#)
- [Aufrufen der Befehlszeilen-Tools](#)
- [Einrichten der Tools](#)
- [Bereitstellung von Anmeldeinformationen für die Tools](#)
- [Umgebungsvariablen](#)

## Voraussetzungen

In diesem Dokument wird davon ausgegangen, dass Sie in einer Linux/ UNIX - oder Windows-Umgebung arbeiten können. Die ElastiCache Amazon-Befehlszeilentools funktionieren auch unter Mac OS X, einer UNIX basierten Umgebung. In diesem Handbuch sind jedoch keine spezifischen Mac OS X-Anweisungen enthalten.

Als Konvention wird jeglichem Befehlszeilentext die generische Eingabeaufforderung **PROMPT>** vorangestellt. Der auf dem Rechner tatsächlich angezeigte Prompt sieht wahrscheinlich anders aus. Wir verwenden auch **\$** , um einen UNIX Linux/spezifischen Befehl und **C:\>** für einen Windows-spezifischen Befehl anzugeben. Die Beispielausgabe des Befehls wird direkt im Anschluss ohne Präfix angezeigt.

## Die Java-Laufzeitumgebung

Für die in diesem Handbuch verwendeten Befehlszeilen-Tools muss Java Version 5 oder höher ausgeführt werden. Entweder eine JRE JDK Oder-Installation ist zulässig. Informationen zum Ansehen und Herunterladen JREs für eine Reihe von Plattformen, einschließlich Linux/ UNIX und Windows, finden Sie unter [Java SE-Downloads](#).

### Einrichten der Java-Home-Variablen

Die Befehlszeilen-Tools sind von einer Umgebungsvariablen (JAVA\_HOME) abhängig, um die Java-Runtime zu finden. Diese Umgebungsvariable sollte auf den vollständigen Pfad des Verzeichnisses gesetzt werden, das ein benanntes Unterverzeichnis enthält bin, das wiederum die ausführbare Datei java (unter Linux und UNIX) oder java.exe (unter Windows) enthält.

### Einrichten der Java Home-Variablen

#### 1. Einrichten der Java Home-Variablen.

- Geben Sie unter Linux UNIX und den folgenden Befehl ein:

```
$ export JAVA_HOME=<PATH>
```

- Geben Sie unter Windows den folgenden Befehl ein:

```
C:\> set JAVA_HOME=<PATH>
```

#### 2. Bestätigen Sie die Pfadeinrichtung, indem Sie `$JAVA_HOME/bin/java -version` ausführen und die Ausgabe überprüfen.

- Unter Linux/ werden Sie eine Ausgabe sehen UNIX, die der folgenden ähnelt:

```
$ $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)
```

- Unter Windows sehen Sie eine Ausgabe vergleichbar der Folgenden:

```
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)
```

## Aufrufen der Befehlszeilen-Tools

Die Befehlszeilentools sind als ZIP Datei auf der [ElastiCache Developer Tools-Website](#) verfügbar. Diese Tools sind in Java geschrieben und enthalten Shell-Skripte für Windows 2000/XP/Vista/Windows 7, Linux/ und Mac. UNIX OSX Die ZIP Datei ist eigenständig und es ist keine Installation erforderlich. Laden Sie einfach die ZIP-Datei herunter und entpacken Sie sie in ein Verzeichnis auf Ihrem lokalen Computer.

## Einrichten der Tools

Die Befehlszeilentools benötigen eine Umgebungsvariable (AWS\_ELASTICACHE\_HOME), um unterstützende Bibliotheken zu finden. Sie müssen diese Umgebungsvariable festlegen, bevor Sie die Tools verwenden können. Setzen Sie sie auf den Pfad des Verzeichnisses, in das Sie die Befehlszeilen-Tools entpackt haben. Dieses Verzeichnis hat den Namen ElastiCacheCli -A.b.nnnn (A, B und n sind Versions-/Release-Nummern) und enthält Unterverzeichnisse mit den Namen bin und lib.

Um die Umgebungsvariable \_\_ festzulegen AWS ELASTICACHE HOME

- Öffnen Sie ein Befehlszeilenfenster und geben Sie einen der folgenden Befehle ein, um die HOME Umgebungsvariable AWS ELASTICACHE \_\_ zu setzen.
  - Geben Sie unter UNIX Linux und den folgenden Befehl ein:

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- Geben Sie unter Windows den folgenden Befehl ein:

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

Um die Verwendung der Tools zu vereinfachen, empfehlen wir Ihnen, das BIN Verzeichnis der Tools zu Ihrem System PATH hinzuzufügen. Im Rest dieses Handbuchs wird davon ausgegangen, dass sich das BIN Verzeichnis in Ihrem Systempfad befindet.

## Um das BIN Verzeichnis der Tools zu Ihrem Systempfad hinzuzufügen

- Geben Sie die folgenden Befehle ein, um das BIN Verzeichnis der Tools zu Ihrem System hinzuzufügen. PATH
  - Geben Sie unter Linux UNIX und den folgenden Befehl ein:

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- Geben Sie unter Windows den folgenden Befehl ein:

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

### Note

Die Windows-Umgebungsvariablen werden zurückgesetzt, wenn Sie das Befehlsfenster schließen. Möglicherweise wollen Sie sie permanent einrichten. Weitere Informationen finden Sie in der Dokumentation zu Ihrer Windows-Version.

### Note

Pfade, die ein Leerzeichen enthalten, müssen in Anführungszeichen gesetzt werden, z. B.: "C:\Program Files\Java"

## Bereitstellung von Anmeldeinformationen für die Tools

Die Befehlszeilentools benötigen den AWS Access Key und den Secret Access Key, die mit Ihrem AWS Konto bereitgestellt wurden. Sie können sie über die Befehlszeile oder aus einer Anmeldeinformationsdatei auf Ihrem lokalen System beziehen.

Die Bereitstellung umfasst eine Vorlagendatei `{AWS_ELASTICACHE_HOME}/credential-file-path.template`, die Sie mit Ihren Informationen bearbeiten müssen. Nachfolgend finden Sie dem Inhalt der Template-Datei:

```
AWS AccessKeyId=<Write your AWS access ID>
```

```
AWS SecretKey=<Write your AWS secret key>
```

### Important

Wenn aktiviertUNIX, beschränken Sie die Berechtigungen auf den Besitzer der Anmeldeinformationsdatei:

```
$ chmod 600 <the file created above>
```

Bei der Einrichtung der Anmeldeinformationsdatei müssen Sie die FILE Umgebungsvariable AWS CREDENTIAL \_\_ festlegen, damit die ElastiCache Tools Ihre Informationen finden können.

Um die FILE Umgebungsvariable AWS CREDENTIAL \_\_ festzulegen

#### 1. Legen Sie die -Umgebungsvariable fest:

- Unter Linux und UNIX aktualisieren Sie die Variable mit dem folgenden Befehl:

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- Legen Sie unter Windows die Variable mit folgendem Befehl fest:

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

#### 2. Überprüfen Sie, ob Ihre Einrichtung ordnungsgemäß funktioniert, und führen Sie den folgenden Befehl aus:

```
elasticache --help
```

Sie sollten die Verwendungsseite für alle ElastiCache Befehle sehen.

## Umgebungsvariablen

Umgebungsvariablen können nützlich sein, um Skripts zu erstellen, Standardwerte zu konfigurieren oder sie vorübergehend zu überschreiben.

Zusätzlich zur FILE Umgebungsvariablen AWS CREDENTIAL \_ \_ unterstützen die meisten in der ElastiCache Befehlszeilenschnittstelle enthaltenen API Tools die folgenden Variablen:

- EC2\_ REGION — Die zu verwendende AWS Region.
- AWS\_ ELASTICACHE \_ URL — Die für den Serviceanruf URL zu verwendende. Es ist nicht erforderlich, einen anderen regionalen Endpunkt anzugeben, wenn EC2 \_ angegeben REGION ist oder der Parameter --region übergeben wird.

Die folgenden Beispiele zeigen, wie die Umgebungsvariable EC2 \_ gesetzt wird REGION, um die von den API Tools verwendete Region zu konfigurieren:

Linux, OS X oder Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

## ElastiCache Amazon-Fehlermeldungen

Die folgenden Fehlermeldungen werden von Amazon zurückgegeben ElastiCache. Möglicherweise erhalten Sie weitere Fehlermeldungen, die von anderen AWS Diensten oder von ElastiCache Valkey, Redis OSS oder Memcached zurückgegeben werden. Beschreibungen von Fehlermeldungen aus anderen Quellen als finden Sie in der Dokumentation der Quelle ElastiCache, die die Fehlermeldung generiert hat.

- [Cluster node quota exceeded](#)
- [Customer's node quota exceeded](#)
- [Manual snapshot quota exceeded](#)
- [Insufficient cache cluster capacity](#)

Fehlermeldung: Das Cluster-Knoten-Kontingent ist erschöpft. Jeder Cluster in dieser Region kann maximal %n Knoten enthalten.

Ursache: Sie haben versucht, einen Cluster zu erstellen oder zu ändern, der dadurch mehr als %n Knoten enthalten würde.

Lösung: Ändern Sie die Anforderung dahingehend, dass der Cluster nicht mehr als %n Knoten enthält. Oder, wenn Sie mehr als %n Knoten benötigen, stellen Sie Ihre Anfrage über das [Amazon ElastiCache Node-Anfrageformular](#).

Weitere Informationen finden Sie unter [ElastiCache Amazon-Limits](#) unter Allgemeine Amazon Web Services-Referenz.

Fehlermeldungen: Kundenknotenquote überschritten. Sie können maximal %n Knoten in dieser Region haben. Oder: Sie haben Ihr Kontingent an %s-Knoten in dieser Region bereits erreicht.

Ursache: Sie haben versucht, einen Cluster zu erstellen oder zu ändern, und Ihr Konto würde dadurch mehr als %n Knoten in allen Clustern in dieser Region enthalten.

Lösung: Ändern Sie die Anforderung dahingehend, dass die Gesamtanzahl der Knoten in der Region für alle Cluster in diesem Konto nicht mehr als &n beträgt. Oder, wenn Sie mehr als %n Knoten benötigen, stellen Sie Ihre Anfrage über das [Amazon ElastiCache Node-Anfrageformular](#).

Weitere Informationen finden Sie unter [ElastiCache Amazon-Limits](#) unter Allgemeine Amazon Web Services-Referenz.

Fehlermeldungen: Die maximale Anzahl an manuellen Snapshots für dieses Cluster innerhalb von 24 Stunden wurde erreicht. oder Die maximale Anzahl an manuellen Snapshots für diesen Konten innerhalb von 24 Stunden für das Kontingent von %n wurde erreicht.

Ursache: Sie haben versucht einen manuellen Snapshot eines Clusters zu erstellen, während die maximale Anzahl an manuellen Snapshots für einen 24-Stunden-Zeitraum bereits erreicht war.

Lösung: Warten Sie 24 Stunden, bevor Sie erneut versuchen, einen manuellen Snapshot des Clusters zu erstellen. Wenn Sie den manuellen Snapshot direkt benötigen, erstellen Sie den Snapshot für einen anderen Konten mit denselben Daten, z. B. für einen anderen Knoten in diesem Cluster.

## Fehlermeldungen: InsufficientCacheClusterCapacity

Ursache: Die verfügbare On-Demand-Kapazität von AWS reicht derzeit nicht aus, um Ihre Anfrage zu bearbeiten.

Solution (Lösung):

- Warten Sie einige Minuten und senden Sie Ihre Anfrage erneut. Die Kapazität kann häufig schwanken.
- Senden Sie eine neue Anfrage mit einer geringeren Anzahl von Knoten oder Shards (Knotengruppen). Wenn Sie z. B. eine einzelne Anfrage zum Starten von 15 Knoten senden möchten, versuchen Sie stattdessen, 3 Anfragen für 5 Knoten oder 15 Anfragen für 1 Knoten zu erstellen.
- Wenn Sie einen Cluster starten, senden Sie eine neue Anfrage ohne Angabe einer Availability Zone.
- Wenn Sie einen Cluster starten, senden Sie eine neue Anfrage unter Verwendung eines anderen Knotentyps (die Skalierung können Sie später erweitern). Weitere Informationen finden Sie unter [Skalierung ElastiCache](#).

## Benachrichtigungen

Dieses Thema behandelt ElastiCache Benachrichtigungen, die Sie interessieren könnten. Eine Benachrichtigung ist eine Situation oder ein Ereignis, die bzw. das in der Regel temporär ist und nur solange andauert, bis eine Lösung gefunden und implementiert wurde. Benachrichtigungen haben normalerweise ein Anfangsdatum und ein Lösungsdatum, nach dem die Benachrichtigung nicht mehr relevant ist. Sie entscheiden selbst, welche Benachrichtigungen für Sie relevant sind. Wir empfehlen eine Implementierungsrichtlinie, durch die sich die Leistung Ihres Clusters verbessern lässt.

Benachrichtigungen kündigen keine neuen oder verbesserten ElastiCache Features oder Funktionen an.

## Allgemeine ElastiCache Benachrichtigungen

Derzeit gibt es keine ausstehenden ElastiCache Benachrichtigungen, die nicht maschinenspezifisch sind.

## ElastiCache (Memcached) Benachrichtigungen

Die folgenden ElastiCache Benachrichtigungen sind spezifisch für die Memcached-Engine.

ElastiCache (Memcached) -spezifische Benachrichtigungen

- [Warnung: Der Memcached-Crawler verursacht Segmentierungsfehler LRU](#)

### Warnung: Der Memcached-Crawler verursacht Segmentierungsfehler LRU

 Datum der Warnung: 28. Februar 2017

Unter bestimmten Umständen kann Ihr Cluster aufgrund eines Segmentierungsfehlers im LRU Memcached Crawler Instabilität aufweisen. Dieses Problem tritt bereits seit einiger Zeit in der Memcached-Engine auf. Das Problem trat in Memcached 1.4.33 zutage, als der Crawler standardmäßig aktiviert war. LRU

Wenn dieses Problem bei Ihnen auftritt, empfehlen wir Ihnen, den LRU Crawler zu deaktivieren, bis es eine Lösung gibt. Führen Sie dazu den Befehl `lru_crawler disable` auf der Befehlszeile aus oder ändern Sie den Parameterwert `lru_crawler` (empfohlen).

Lösungsdatum:

Auflösung

## ElastiCache (RedisOSS) spezifische Benachrichtigungen

Derzeit gibt es keine ausstehenden ElastiCache (Redis-OSS) Benachrichtigungen.

# ElastiCache Geschichte der Dokumentation

- APIVersion: 2015-02-02
- Letzte Aktualisierung der Dokumentation: 27. November 2023

In der folgenden Tabelle werden wichtige Änderungen in den einzelnen Versionen des ElastiCache Benutzerhandbuchs nach März 2018 beschrieben. Wenn Sie über Aktualisierungen dieser Dokumentation informiert werden möchten, können Sie den RSS Feed abonnieren.

## Aktuelle ElastiCache Aktualisierungen

Änderung	Beschreibung	Datum
<a href="#">Support für ElastiCache mit Valkey</a>	ElastiCache unterstützt jetzt Valkey. <a href="#">Valkey 7.2.6 ist mit Redis OSS 7.2 kompatibel.</a> <a href="#">Weitere Informationen finden Sie unter Valkey.</a>	8. Oktober 2024
<a href="#">Größe: flexible reservierte Knoten</a>	ElastiCache unterstützt jetzt <a href="#">Size flexible reservierte Knoten</a> . Weitere Informationen finden Sie unter <a href="#">ElastiCache Amazon-Preise</a> .	1. Oktober 2024
<a href="#">ElastiCache (RedisOSS) hat Unterstützung für zusätzliche C7gn-Knotengrößen hinzugefügt</a>	ElastiCache (RedisOSS) fügte Unterstützung für zusätzliche C7gn-Knotengrößen hinzu.	10. Januar 2024
<a href="#">ElastiCache (RedisOSS) unterstützt jetzt die Erstellung von serverlosen Caches</a>	Sie können jetzt Serverless-Caches erstellen, die das Cache-Management vereinfachen und sofort skalieren, um die anspruchsvollsten Anwendungen zu unterstützen. Weitere	8. November 2023

Informationen finden Sie unter [Auswählen aus den Bereitstellungsoptionen](#). Im Rahmen dieser Funktion wurden [neue Berechtigungen](#) hinzugefügt, um ElastiCacheServiceRolePolicy die Zuordnung von serverlosen AmazonElastiCacheFullAccess Caches zu verwalteten Endpunkten zu ermöglichen. VPC Darüber hinaus wurden Berechtigungen hinzugefügt, um ein überarbeitetes Konsolenerlebnis mithilfe der Richtlinie AmazonElastiCacheFullAccess zu unterstützen.

[ElastiCache \(Memcached\) unterstützt jetzt die Erstellung von serverlosen Caches](#)

Sie können jetzt Serverless-Caches erstellen, die das Cache-Management vereinfachen und sofort skalieren, um die anspruchsvollsten Anwendungen zu unterstützen. Weitere Informationen finden Sie unter [Auswählen aus den Bereitstellungsoptionen](#). Im Rahmen dieser Funktion wurden [neue Berechtigungen](#) hinzugefügt, um die Zuordnung von serverlosen ElastiCacheServiceRolePolicy Caches AmazonElastiCacheFullAccess zu verwalteten Endpunkten zu ermöglichen. VPC Darüber hinaus wurden Berechtigungen hinzugefügt, um ein überarbeitetes Konsolenerlebnis mithilfe der Richtlinie AmazonElastiCacheFullAccess zu unterstützen.

8. November 2023

[ElastiCache \(RedisOSS\) unterstützt jetzt das Ändern des Clustermodus](#)

Sie können jetzt Cluster vom Clustermodus Deaktiviert (CMD) in den Clustermodus Aktiviert (CME) migrieren. Weitere Informationen finden Sie unter [Ändern des Clustermodus](#).

11. Mai 2023

[ElastiCache \(RedisOSS\) unterstützt jetzt das Ändern der Verschlüsselungseinstellungen bei der Übertragung](#)

Sie können jetzt die TLS Konfiguration Ihrer OSS Redis-Cluster ändern, ohne die Cluster neu erstellen oder bereitstellen zu müssen oder die Anwendungsverfügbarkeit zu beeinträchtigen. Weitere Informationen finden Sie unter [Aktivieren der Verschlüsselung während der Übertragung für einen vorhandenen Cluster](#).

28. Dezember 2022

[ElastiCache \(RedisOSS\) unterstützt jetzt die Authentifizierung von Benutzern mit IAM](#)

IAM Die Authentifizierung ermöglicht es Ihnen, eine Verbindung zu ElastiCache (RedisOSS) mithilfe von Identitäten zu authentifizieren. AWS IAM Auf diese Weise können Sie Ihr Sicherheitsmodell stärken und viele administrative Sicherheitsaufgaben vereinfachen. [Weitere Informationen finden Sie unter Authentifizierung mit IAM](#)

16. November 2022

[ElastiCache \(RedisOSS\) unterstützt jetzt OSS Redis 7](#)

Diese Version bietet Amazon ElastiCache (RedisOSS) mehrere neue Funktionen: OSS Redis-Funktionen, ACL Verbesserungen und Sharded Pub/Sub. Weitere Informationen finden Sie unter [ElastiCache \(Redis\) Version 7.0. OSS](#)

08. November 2022

## [ElastiCache \(RedisOSS\) unterstützt jetzt IPv6](#)

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er nur IPv4 Verbindungen, nur Verbindungen oder beides IPv4 und IPv6 IPv6 Verbindungen (Dual-Stack) akzeptiert. IPv6 [wird für Workloads unterstützt, die die OSS Redis-Engine ab Version 6.2 auf allen auf dem Nitro-System basierenden Instanzen verwenden](#). Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6 Weitere Informationen zu Netzwerktypen finden Sie unter [Auswählen eines Netzwerktyps](#).

07. November 2022

## [ElastiCache \(Memcached\) unterstützt jetzt IPv6](#)

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er nur IPv4 Verbindungen, nur Verbindungen oder beides IPv4 und IPv6 IPv6 Verbindungen (Dual-Stack) akzeptiert. [IPv6 wird für Workloads unterstützt, die die Memcached Engine ab Version 1.6.6 auf allen auf dem Nitro-System basierenden Instanzen verwenden.](#) Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6 Weitere Informationen zu Netzwerktypen finden Sie unter [Auswählen eines Netzwerktyps](#).

07. November 2022

[ElastiCache \(Memcached\) unterstützt jetzt die Verschlüsselung bei der Übertragung](#)

Die Verschlüsselung während der Übertragung ist eine optionale Funktion, mit der Sie die Sicherheit Ihrer Daten an ihrer verwundbarsten Stelle verbessern können – bei der Übertragung von einem Speicherort an einen anderen. Sie wird in Memcached-Versionen 1.6.12 und höher unterstützt. Weitere Informationen finden Sie unter [Verschlüsselung bei der ElastiCache Übertragung](#) (). TLS

26. Mai 2022

[ElastiCache \(RedisOSS\) unterstützt jetzt das native Format JavaScript Object Notation \(JSON\)](#)

Das native Format JavaScript Object Notation (JSON) ist eine einfache, schemalose Methode, komplexe Datensätze innerhalb von Redis-Clustern zu kodieren. OSS Sie können Daten mithilfe des JavaScript Object Notation (JSON) - Formats nativ innerhalb von OSS Redis-Clustern speichern und darauf zugreifen und die in diesen Clustern gespeicherten JSON Daten aktualisieren, ohne dass Sie benutzerdefinierten Code für die Serialisierung und Deserialisierung verwalten müssen. [Weitere Informationen finden Sie unter Erste Schritte mit. JSON](#)

25. Mai 2022

## [ElastiCache unterstützt jetzt PrivateLink](#)

AWS PrivateLink ermöglicht Ihnen den privaten Zugriff auf ElastiCache API Vorgänge ohne Internet-Gateway, NAT Gerät, VPN Verbindung oder AWS Direct Connect-Verbindung. Weitere Informationen finden Sie unter [Amazon ElastiCache API und VPC Schnittstellenendpunkte \(AWS PrivateLink\)](#) für Redis OSS oder [Amazon ElastiCache API und VPC Schnittstellenendpunkte \(AWS PrivateLink\)](#) für Memcached.

24. Januar 2022

[ElastiCache \(RedisOSS\) unterstützt jetzt Redis 6.2 und Data Tiering OSS](#)

Amazon ElastiCache (RedisOSS) stellt die nächste Version der OSS Redis-Engine vor, die von Amazon unterstützt wird. ElastiCache ElastiCache (RedisOSS) 6.2 beinhaltet Leistungsverbesserungen für TLS-fähige Cluster, die x86-Knotentypen mit 8 vCPUs oder mehr oder Graviton2-Knotentypen mit 4 oder mehr verwenden. vCPUs ElastiCache (RedisOSS) führt auch Daten-Tiering ein. Sie können Daten-Tiering als kostengünstigere Methode verwenden, um Ihre Cluster auf bis zu Hunderte von Terabyte Kapazität zu skalieren. [Weitere Informationen finden Sie unter ElastiCache \(RedisOSS\) Version 6.2 \(erweitert\) und Data Tiering.](#)

23. November 2021

## [Support für Auto Scaling](#)

ElastiCache (RedisOSS) unterstützt jetzt Auto Scaling. ElastiCache (RedisOSS) Auto Scaling ist die Fähigkeit, die gewünschten Shards oder Replikate in Ihrem ElastiCache (Redis OSS -) Dienst automatisch zu erhöhen oder zu verringern. ElastiCache nutzt den Application Auto Scaling Scaling-Dienst, um diese Funktionalität bereitzustellen. Weitere Informationen finden Sie unter [Auto Scaling ElastiCache \(RedisOSS\) - Cluster](#).

19. August 2021

## [Support für die Lieferung von Redis OSS Slow-Logs](#)

ElastiCache ermöglicht es Ihnen jetzt, Redis OSS SLOWLOG an eines von zwei Zielen zu streamen: Amazon Data Firehose oder Amazon CloudWatch Logs. Weitere Informationen finden Sie unter [Protokollauslieferung](#).

22. April 2021

[Support für die Markierung von Ressourcen und Bedingungsschlüsseln](#)

ElastiCache unterstützt jetzt Tagging, um Sie bei der Verwaltung Ihrer Cluster und anderer Ressourcen zu unterstützen. ElastiCache Weitere Informationen finden Sie unter [Taggen Ihrer ElastiCache Ressource](#). ElastiCache führt auch Unterstützung für Bedingungsschlüssel ein. Sie können Bedingungen angeben, die bestimmen, wie eine IAM Richtlinie wirksam wird. Weitere Informationen finden Sie unter [Verwenden von Bedingungsschlüssel](#).

7. April 2021

[Support für die Markierung von Ressourcen und Bedingungsschlüsseln](#)

ElastiCache unterstützt jetzt Tagging, um Sie bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen zu unterstützen. Weitere Informationen finden Sie unter [Taggen Ihrer ElastiCache Ressource](#). ElastiCache führt auch Unterstützung für Bedingungsschlüssel ein. Sie können Bedingungen angeben, die bestimmen, wie eine IAM Richtlinie wirksam wird. Weitere Informationen finden Sie unter [Verwenden von Bedingungsschlüssel](#).

7. April 2021

## [ElastiCache ist jetzt auf AWS Outposts verfügbar](#)

[AWS Outposts](#) bieten native AWS Dienste, Infrastrukturen und Betriebsmodelle für praktisch jedes Rechenzentrum, jeden Colocation-Bereich oder jede lokale Einrichtung. Sie können ElastiCache auf Outposts bereitstellen, um Cache vor Ort einzurichten, zu betreiben und zu verwenden, genau wie in der Cloud. Weitere Informationen finden Sie unter [Outposts for Redis verwenden OSS oder Outposts for Memcached verwenden](#).

8. Oktober 2020

## [ElastiCache OSSunterstützt jetzt Redis 6](#)

Amazon ElastiCache (RedisOSS) stellt die nächste Version der OSS Redis-Engine vor, die von Amazon unterstützt wird. ElastiCache Diese Version umfasst die [Authentifizierung von Benutzern mit rollenbasierter Zugriffskontrolle](#), versionlosen Support, clientseitiges Caching und erhebliche betriebliche Verbesserungen. Weitere Informationen finden Sie unter [ElastiCache \(RedisOSS\) Version 6.0 \(erweitert\)](#).

7. Oktober 2020

## [ElastiCache unterstützt jetzt Local Zones](#)

Eine lokale Zone ist eine Erweiterung einer AWS Region, die sich geografisch in der Nähe Ihrer Benutzer befindet. Sie können jede virtuelle private Cloud (VPC) von einer übergeordneten AWS Region auf Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es einer lokalen Zone zuweisen. Weitere Informationen finden Sie unter [Local Zones nutzen](#).

25. September 2020

## [ElastiCache \(RedisOSS\) unterstützt jetzt die Skalierung Ihrer OSS Redis-Cluster-Umgebung auf bis zu 500 Knoten oder 500 Shards](#)

Der OSS Redis-Cluster-Modus ermöglicht Konfigurationen, mit denen Sie Ihre Daten auf mehrere Shards partitionieren können, und bietet eine bessere Skalierbarkeit, Leistung und Verfügbarkeit. Diese Funktion ist ab Amazon ElastiCache (RedisOSS) Version 5.0.6 in allen AWS Regionen und für alle vorhandenen und neuen ElastiCache (RedisOSS) Cluster-Umgebungen verfügbar. Weitere Informationen finden Sie unter [Redis OSS Nodes and Shards](#).

13. August 2020

[ElastiCache unterstützt jetzt Berechtigungen auf Ressourceebene](#)

Sie können jetzt den Umfang der Benutzerberechtigungen einschränken, indem Sie ElastiCache Ressourcen in einer AWS Identity and Access Management (IAM) - Richtlinie angeben. Weitere Informationen finden Sie unter [Berechtigungen auf Ressourceebene](#).

12. August 2020

[ElastiCache \(RedisOSS\) fügt zusätzliche CloudWatch Amazon-Metriken hinzu](#)

ElastiCache (RedisOSS) unterstützt jetzt neue CloudWatch Metriken, darunter PubSubCmds und HyperLogLogBasedCmds. Eine vollständige Liste finden Sie unter [Metrics for OSS Redis](#).

10. Juni 2020

[ElastiCache unterstützt jetzt die automatische Aktualisierung von Clustern ElastiCache](#)

Amazon unterstützt ElastiCache jetzt die automatische Aktualisierung von ElastiCache Clustern, nachdem das „empfohlene Verfallsdatum“ des Service-Updates abgelaufen ist. ElastiCache verwendet Ihr Wartungsfenster, um die automatische Aktualisierung der entsprechenden Cluster zu planen. Weitere Informationen finden Sie unter [Self-Service-Updates](#).

13. Mai 2020

[ElastiCache \(RedisOSS\)  
\) unterstützt jetzt Global  
Datastore für Redis OSS](#)

Die OSS Funktion Global Datastore for Redis bietet eine vollständig verwaltete, schnelle, zuverlässige und sichere Replikation über Regionen hinweg. AWS Mit dieser Funktion können Sie regionsübergreifende Read Replica-Cluster für ElastiCache (RedisOSS) erstellen, um Lesevorgänge mit geringer Latenz und regionsübergreifende Disaster Recovery zu ermöglichen. AWS Sie können einen globalen Datenspeicher erstellen, ändern und beschreiben. Sie können auch AWS Regionen zu Ihrem globalen Datenspeicher hinzufügen oder daraus entfernen und eine Region innerhalb eines globalen Datenspeichers als primäre AWS Region heraufstufen. Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation](#) mithilfe von Global Datastore.

16. März 2020

[ElastiCache \(RedisOSS\)  
unterstützt jetzt Redis Version  
5.0.6 OSS](#)

Weitere Informationen finden Sie unter [ElastiCache \(RedisOSS\) Version 5.0.6 \(erweitert\)](#).

18. Dezember 2019

## [Amazon unterstützt ElastiCache jetzt T3-Standard-Cache-Knoten](#)

Sie können jetzt die Allzweck-Burstable-T3-Standard-Cache-Knoten der nächsten Generation in Amazon starten. ElastiCache EC2Die T3-Standard-Instances von Amazon bieten ein grundlegendes CPU Leistungsniveau mit der Möglichkeit, die CPU Nutzung jederzeit zu beschleunigen, bis das aufgelaufene Guthaben aufgebraucht ist. Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

12. November 2019

[Amazon unterstützt ElastiCache jetzt die Änderung des AUTH Tokens auf einem vorhandenen ElastiCache \(Redis-OSS\) Server](#)

ElastiCache (RedisOSS) 5.0.6 ermöglicht es Ihnen jetzt, Authentifizierungstoken zu ändern, indem Sie neue Token setzen und rotieren. Sie können aktive Token jetzt ändern, während sie in Gebrauch sind. Sie können vorhandenen Clustern mit aktivierter Verschlüsselung während der Übertragung, die zuvor ohne Authentifizierungstoken eingerichtet wurden, auch ganz neue Token hinzufügen. Dies ist ein zweistufiger Prozess, mit dem Sie das Token festlegen und rotieren können, ohne die Client-Anforderungen zu unterbrechen. Diese Funktion wird derzeit nicht unterstützt auf. AWS CloudFormation Weitere Informationen finden Sie unter [Benutzer mit dem OSS AUTH Redis-Befehl authentifizieren](#).

30. Oktober 2019

[Amazon unterstützt ElastiCache jetzt die Online-Datenmigration von Redis OSS auf Amazon EC2](#)

Sie können jetzt Online Migration verwenden, um Ihre Daten von selbst gehostetem Redis OSS auf Amazon EC2 zu Amazon zu migrieren. ElastiCache Weitere Informationen finden Sie unter [Online-Migration](#) zu. ElastiCache

28. Oktober 2019

[ElastiCache \(RedisOSS\) führt vertikale Online-Skalierung für den Redis-Clustermodus ein. OSS](#)

Sie können Ihren OSS Sharded-Redis-Cluster jetzt bei Bedarf hoch- oder herunterskalieren. ElastiCache (RedisOSS) passt die Größe Ihres Clusters an, indem es den Knotentyp ändert, während der Cluster weiterhin online bleibt und eingehende Anfragen bearbeitet. Weitere Informationen finden Sie unter [Vertikales Online-Skalieren durch Ändern des Knotentyps](#).

20. August 2019

[ElastiCache \(RedisOSS\) ermöglicht es Benutzern jetzt, einen einzigen Reader-Endpoint für Ihren Amazon ElastiCache \(RedisOSS\) -Cluster zu verwenden.](#)

Mit dieser Funktion können Sie den gesamten Lesetraffic über einen einzigen Endpunkt auf Clusterebene an Ihren ElastiCache (RedisOSS) -Cluster weiterleiten, um den Lastenausgleich und die höhere Verfügbarkeit zu nutzen. Weitere Informationen finden Sie unter [Finden von Verbindungsendpunkten](#).

13. Juni 2019

[ElastiCache \(RedisOSS\) ermöglicht es Benutzern jetzt, Dienstupdates nach ihrem eigenen Zeitplan anzuwenden](#)

Dank dieser Funktion können Sie verfügbare Service-Updates zu einem Zeitpunkt Ihrer Wahl anwenden, nicht nur in Wartungsfenstern. Dadurch werden Dienstunterbrechungen minimiert, insbesondere bei hohen Geschäftsabläufen, und Sie können sicherstellen, dass Sie die Vorschriften einhalten, wenn Ihr Cluster an ElastiCache unterstützten Compliance-Programmen teilnimmt. Weitere Informationen finden Sie unter [Self-Service-Updates bei Amazon ElastiCache](#) und [Compliance-Validierung für Amazon ElastiCache](#).

4. Juni 2019

[ElastiCache Standardangebote für Reserved Instances: Teilweise Vorauszahlung, vollständige Vorauszahlung und keine Vorauszahlung.](#)

Reserved Instances bieten Ihnen die Flexibilität, je nach ElastiCache Instance-Typ und AWS Region eine Amazon-Instance für einen Zeitraum von einem oder drei Jahren zu reservieren. Weitere Informationen finden Sie unter [Verwalten der Kosten mit reservierten Knoten](#).

18. Januar 2019

[ElastiCache \(Redis-OSS\)  
Unterstützung für bis zu 250  
Knoten pro Redis-Cluster OSS](#)

Das Knoten- oder Shard-Limit kann auf maximal 250 pro ElastiCache (OSSRedis-) Cluster erhöht werden. Weitere Informationen finden Sie unter [Shards](#).

19. November 2018

[ElastiCache \(RedisOSS\) -  
Unterstützung für Autofailo-  
ver sowie Sicherung und  
Wiederherstellung auf allen  
T2-Knoten](#)

ElastiCache (RedisOSS) bietet Unterstützung für Autofailover, Erstellung von Snapshots sowie Sicherung und Wiederherstellung auf allen T2-Knoten. Weitere Informationen finden Sie unter [ElastiCache \(RedisOSS\)  
Backup und Wiederherstellung  
und Snapshot](#).

19. November 2018

[ElastiCache \(RedisOSS\) -  
Unterstützung für M5- und R5-  
Knoten](#)

ElastiCache (RedisOSS) unterstützt jetzt M5- und R5-Knoten, allgemeine und speicheroptimierte Instance-Typen, die auf dem Nitro-System basieren. AWS Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

23. Oktober 2018

[Unterstützung für eine dynamische Änderung der Anzahl von Read Replicas](#)

ElastiCache (RedisOSS) bietet zusätzliche Unterstützung für das Hinzufügen und Entfernen von Read Replicas aus beliebigen Clustern ohne Cluster-Ausfallzeiten. Weitere Informationen zu diesen und anderen Änderungen in dieser Version finden Sie unter [Ändern der Anzahl der Replikate](#) im ElastiCache (Redis OSS -) Benutzerhandbuch. Siehe auch [DecreaseReplicaCount](#) und [IncreaseReplicaCount](#) in der ElastiCache API Referenz.

17. September 2018

[RAMP Konformitätsbescheinigung der Fed](#)

ElastiCache (RedisOSS) ist jetzt für die RAMP Einhaltung der Vorschriften durch die Fed zertifiziert. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

30. August 2018

[Valkey- oder OSS Redis-Engine-Upgrades \(Cluster-Modus aktiviert\)](#)

Amazon ElastiCache (RedisOSS) hat Unterstützung für das Upgrade von Valkey- oder Redis-Engine-Versionen OSS (Cluster-Modus aktiviert) hinzugefügt. Weitere Informationen finden Sie unter [Upgrade von Engine-Versionen](#).

20. August 2018

## [PCIDSSKonformitätszertifizierung](#)

ElastiCache (RedisOSS) ist jetzt für die PCI DSS Einhaltung der Vorschriften zertifiziert. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

5. Juli 2018

## [Support für ElastiCache \(RedisOSS\) 4.0.10](#)

ElastiCache (RedisOSS) unterstützt jetzt Redis OSS 4.0.10, einschließlich Verschlüsselung und Online-Clustergrößenänderung in einer einzigen Version. Weitere Informationen finden Sie unter [ElastiCache \(RedisOSS\) Version 4.0.10 \(erweitert\)](#).

14. Juni 2018

## [Neu strukturiertes Benutzerhandbuch](#)

[Das einheitliche ElastiCache Benutzerhandbuch ist jetzt neu strukturiert, sodass es separate Benutzerhandbücher für Redis \(\(Redis\) User Guide\) und für Memcached OSS ElastiCache \(\(MemcachedOSS\) User Guide\) gibt.](#) [ElastiCache](#) Die Dokumentationsstruktur im Abschnitt [AWS CLI Command Reference : Elasticache](#) und [Amazon ElastiCache API Reference](#) bleibt unverändert.

20. April 2018

## [Support für ngineCPUUtilization E-Metrik](#)

ElastiCache (RedisOSS) hat eine neue Metrik hinzugefügt `EngineCPUUtilization`, die den Prozentsatz Ihrer Kapazität CPU angibt, der derzeit genutzt wird. Weitere Informationen finden Sie unter [Metriken für OSS Redis](#).

9. April 2018

In der folgenden Tabelle werden die wichtigen Änderungen am ElastiCache Benutzerhandbuch vor März 2018 beschrieben.

Änderung	Beschreibung	Änderungsdatum
Support für die Region Asien-Pazifik (Osaka – regional).	<p>ElastiCache Unterstützung für die Region Asien-Pazifik (lokal in Osaka) hinzugefügt. Die Region Asien-Pazifik (Osaka) unterstützt derzeit eine einzige Availability Zone und ist nur auf Einladung zugänglich. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> <li>• <a href="#">Unterstützte Regionen</a></li> <li>• <a href="#">Unterstützte Cache-Knotentypen</a></li> </ul>	12. Februar 2018
Support für die Europa (Paris).	<p>ElastiCache Unterstützung für die Region EU (Paris) hinzugefügt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> <li>• <a href="#">Unterstützte Regionen</a></li> <li>• <a href="#">Unterstützte Cache-Knotentypen</a></li> </ul>	18. Dezember 2017
Support für die Region China (Ningxia)	<p>Amazon ElastiCache hat Unterstützung für die Region China (Ningxia) hinzugefügt. Weitere Informationen finden Sie hier:</p>	11. Dezember 2017

Änderung	Beschreibung	Änderungsdatum
	<ul style="list-style-type: none"> <li>• <a href="#">Unterstützte Regionen</a></li> <li>• <a href="#">Unterstützte Cache-Knotentypen</a></li> </ul>	
Unterstützung für dienstverknüpfte Rollen	<p>In dieser Version ElastiCache wurde Unterstützung für Service Linked Roles (SLR) hinzugefügt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> <li>• <a href="#">Verwenden von serviceverknüpften Rollen für Amazon ElastiCache</a></li> <li>• <a href="#">Richten Sie Ihre Berechtigungen ein (nur ElastiCache für neue Benutzer)</a></li> </ul>	7. Dezember 2017
Unterstützung für R4-Knotentypen	<p>Diese Version von ElastiCache hat Unterstützung für R4-Knotentypen in allen AWS Regionen hinzugefügt, die von ElastiCache unterstützt werden. R4-Knotentypen sind als On-Demand-Knoten oder als reservierte Cache-Knoten erhältlich. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> <li>• <a href="#">Unterstützte Cache-Knotentypen</a></li> <li>• <a href="#">Spezifische Parameter des Memcached-Knotentyps</a></li> <li>• <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a></li> </ul>	20. November 2017

Änderung	Beschreibung	Änderungsdatum
ElastiCache (RedisOSS) 3.2.10 und Unterstützung für Online-Resharding	<p>Amazon ElastiCache (RedisOSS) fügt Unterstützung für ElastiCache (RedisOSS) 3.2.10 hinzu. ElastiCache (RedisOSS) führt außerdem die Online-Clustergrößenänderung ein, um Shards zum Cluster hinzuzufügen oder daraus zu entfernen, während der Cluster weiterhin eingehende I/O-Anfragen bearbeitet. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Online-Größenanpassung von Clustern</a></li><li>• <a href="#">Online-Resharding für Valkey oder OSS Redis (Clustermodus aktiviert)</a></li></ul>	9. November 2017
HIPAAEignung	<p>ElastiCache (RedisOSS) Version 3.2.6 ist jetzt für die HIPAA Eignung zertifiziert, wenn die Verschlüsselung auf Ihrem Cluster aktiviert ist. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Konformitätsvalidierung für Amazon ElastiCache</a></li><li>• <a href="#">Datensicherheit bei Amazon ElastiCache</a></li></ul>	2. November 2017

Änderung	Beschreibung	Änderungsdatum
ElastiCache (RedisOSS) 3.2.6 und Unterstützung für Verschlüsselung	<p>ElastiCache fügt Unterstützung für ElastiCache (RedisOSS) 3.2.6 hinzu, die zwei Verschlüsselungsfunktionen beinhaltet:</p> <ul style="list-style-type: none"><li>• Die Verschlüsselung der Daten während der Übertragung verschlüsselt Ihre Daten bei jeder Übertragung, z. B. zwischen Knoten in einem Cluster oder zwischen einem Cluster und Ihrer Anwendung.</li><li>• Bei der Verschlüsselung im Ruhezustand werden Ihre Daten auf dem Laufwerk bei Synchronisierungs- und Sicherungsoperationen verschlüsselt.</li></ul> <p>Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Datensicherheit bei Amazon ElastiCache</a></li><li>• <a href="#">Unterstützte Engines und Versionen</a></li></ul>	25. Oktober 2017
Thema der Verbindungsmuster	<p>ElastiCache Die Dokumentation fügt ein Thema hinzu, das verschiedene Muster für den Zugriff auf einen ElastiCache Cluster in einem Amazon VPC behandelt.</p> <p>Weitere Informationen finden Sie <a href="#">Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einem Amazon VPC</a> im ElastiCache Benutzerhandbuch.</p>	24. April 2017

Änderung	Beschreibung	Änderungsdatum
Support für Memcached 1.4.34	<p>ElastiCache fügt Unterstützung für Memcached Version 1.4.34 hinzu, die eine Reihe von Korrekturen für frühere Memcached-Versionen beinhaltet.</p> <p>Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.4.34 auf <a href="#">Memcached</a> am. GitHub</p>	10. April 2017
Unterstützung für das Testen des automatischen Failovers	<p>ElastiCache fügt Unterstützung für das Testen von Automatic Failover auf Redis-Clustern hinzu, die Replikation unterstützen. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Testen des automatischen Failovers</a> im ElastiCache -Benutzerhandbuch.</li><li>• <a href="#">TestFailover</a> in der Referenz. ElastiCache API</li><li>• <a href="#">test-failover</a> in der AWS CLI -Referenz.</li></ul>	4. April 2017
Verbesserte Redis-Wiederherstellung OSS	<p>ElastiCache fügt erweiterte OSS Redis-Backup- und Wiederherstellungsfunktionen mit Cluster-Größenänderung hinzu. Diese Funktion unterstützt die Wiederherstellung einer Sicherung in einem Cluster mit einer unterschiedlichen Anzahl an Shards als bei dem Cluster, der zur Erstellung der Sicherung verwendet wurde. (Für den API und CLI kann diese Funktion eine andere Anzahl von Knotengruppen als eine andere Anzahl von Shards wiederherstellen.) Dieses Update unterstützt auch verschiedene OSS Redis-Steckplatzkonfigurationen. Weitere Informationen finden Sie unter <a href="#">Wiederherstellen aus einem Backup in einen neuen Cache</a>.</p>	15. März 2017

Änderung	Beschreibung	Änderungsdatum
Neuer OSS Redis-Speicherverwaltungsparmeter	<p>ElastiCache fügt einen neuen OSS Redis-Parameter <code>reserved-memory-percent</code> hinzu, der die Verwaltung Ihres reservierten Speichers erleichtert. Dieser Parameter ist in allen Versionen von ElastiCache (RedisOSS) verfügbar. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Verwaltung des reservierten Speichers für Valkey und Redis OSS</a></li><li>• <a href="#">Neue Parameter für Redis 3.2.4 OSS</a></li></ul>	15. März 2017
Unterstützung für Memcached 1.4.33	<p>ElastiCache fügt Unterstützung für Memcached Version 1.4.33 hinzu. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Memcached-Version 1.4.33</a></li><li>• <a href="#">In Memcached 1.4.33 hinzugefügte Parameter</a></li></ul>	20. Dezember 2016
Unterstützung für die Region EU (London)	<p>ElastiCache fügt Unterstützung für die Region EU (London) hinzu. Derzeit werden nur die Knotentypen T2 und M4 unterstützt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Unterstützte Regionen</a></li><li>• <a href="#">Unterstützte Cache-Knotentypen</a></li></ul>	13. Dezember 2016
Unterstützung für die Region Kanada (Montreal)	<p>ElastiCache fügt Unterstützung für die Region Kanada (Montreal) hinzu. In dieser AWS Region werden derzeit nur die Knotentypen M4 und T2 unterstützt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Unterstützte Regionen</a></li><li>• <a href="#">Unterstützte Cache-Knotentypen</a></li></ul>	8. Dezember 2016

Änderung	Beschreibung	Änderungsdatum
Unterstützung für Knotentypen M4 und R3	<p>ElastiCache fügt Unterstützung für die Knotentypen R3 und M4 in der Region Südamerika (São Paulo) und die Knotentypen M4 in der Region China (Peking) hinzu. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Unterstützte Regionen</a></li><li>• <a href="#">Unterstützte Cache-Knotentypen</a></li></ul>	1. November 2016
Unterstützung für die Region USA Ost 2 (Ohio)	<p>ElastiCache fügt Unterstützung für die Region USA Ost (Ohio) (us-east-2) mit den Knotentypen M4, T2 und R3 hinzu. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Unterstützte Regionen</a></li><li>• <a href="#">Unterstützte Cache-Knotentypen</a></li></ul>	17. Oktober 2016

Änderung	Beschreibung	Änderungsdatum
Support für Redis Cluster OSS	<p>ElastiCache fügt Unterstützung für Redis OSS Cluster hinzu (erweitert). Kunden, die Redis OSS Cluster verwenden, können ihre Daten auf bis zu 15 Shards (Knotengruppen) partitionieren. Jeder Shard unterstützt Replikation mit bis zu 5 Leserepliken pro Shard. Die automatischen Failover-Zeiten von Redis OSS Cluster sind etwa ein Viertel so lang wie bei früheren Versionen.</p> <p>Diese Version umfasst eine neu gestaltete Verwaltungskonsole, die Terminologie entsprechend der Branchennutzung verwendet.</p> <p>Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Vergleich von Memcached und Redis OSS</a></li><li>• <a href="#">ElastiCache Komponenten und Funktionen</a> – Beachten Sie die Abschnitte zu Knoten, Shards, Clustern und Replikation.</li><li>• <a href="#">ElastiCache Terminologie</a></li></ul>	12. Oktober 2016
Unterstützung für Knotentyp M4	<p>ElastiCache fügt Unterstützung für die M4-Familie von Knotentypen in den meisten AWS Regionen hinzu, die von unterstützt werden. ElastiCache M4-Knotentypen sind als On-Demand-Knoten oder als reservierte Cache-Knoten erhältlich. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"><li>• <a href="#">Unterstützte Cache-Knotentypen</a></li><li>• <a href="#">Spezifische Parameter des Memcached-Knotentyps</a></li><li>• <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a></li></ul>	3. August 2016

Änderung	Beschreibung	Änderungsdatum
Unterstützung der Region Mumbai	<p>ElastiCache fügt Unterstützung für die Region Asien-Pazifik (Mumbai) hinzu. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> <li>• <a href="#">Unterstützte Cache-Knotentypen</a></li> <li>• <a href="#">Spezifische Parameter des Memcached-Knotentyps</a></li> <li>• <a href="#">OSSRedis-Knotentyp-spezifische Parameter</a></li> </ul>	27. Juni 2016
Snapshot-Export	<p>ElastiCache fügt die Möglichkeit hinzu, einen OSS Redis-Snapshot zu exportieren, sodass Sie von außen ElastiCache darauf zugreifen können. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> <li>• <a href="#">Exportieren einer Sicherung</a> im ElastiCache Amazon-Benutzerhandbuch</li> <li>• <a href="#">CopySnapshot</a> in der ElastiCache API Amazon-Referenz</li> </ul>	26. Mai 2016
Vertikale Skalierung von Knotentypen	<p>ElastiCache fügt die Möglichkeit hinzu, Ihren OSS Redis-Knotentyp zu skalieren. Weitere Informationen finden Sie unter <a href="#">Skalierung ElastiCache</a>.</p>	24. März 2016
Einfaches Upgraden der Engine	<p>ElastiCache fügt die Möglichkeit hinzu, Ihre OSS Redis-Cache-Engine einfach zu aktualisieren. Weitere Informationen finden Sie unter <a href="#">Versionsverwaltung für ElastiCache</a>.</p>	22. März 2016
Unterstützung für R3-Knotentypen	<p>ElastiCache fügt Unterstützung für R3-Knotentypen in den Regionen China (Peking) und Südamerika (São Paulo) hinzu. Weitere Informationen finden Sie unter <a href="#">Unterstützte Cache-Knotentypen</a>.</p>	16. März 2016

Änderung	Beschreibung	Änderungsdatum
Zugreifen ElastiCache mit einer Lambda-Funktion	Es wurde ein Tutorial zur Konfiguration einer Lambda-Funktion für den Zugriff ElastiCache in einem Amazon VPC hinzugefügt. Weitere Informationen finden Sie unter <a href="#">Weitere ElastiCache Tutorials und Videos</a> .	12. Februar 2016
Support für Redis 2.8.24 OSS	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.24 hinzu, wobei Verbesserungen seit Redis 2.8.23 hinzugefügt wurden. OSS Zu den Verbesserungen gehören Fehlerbehebungen und die Unterstützung einer Protokollierung von Adressen ungültiger Speicherzugriffe. Weitere Informationen finden Sie hier: <ul style="list-style-type: none"><li>• <a href="#">ElastiCache (RedisOSS) Version 2.8.24 (erweitert)</a></li><li>• <a href="#">Versionshinweise zu Redis 2.8 OSS</a></li></ul>	20. Januar 2016
Support für die Region Asien-Pazifik (Seoul)	ElastiCache fügt Unterstützung für die Region Asien-Pazifik (Seoul) (ap-northeast-2) mit den Knotentypen t2, m3 und r3 hinzu.	6. Januar 2016
Änderung der ElastiCache Amazon-Konsole.	Da die neueren OSS Redis-Versionen eine bessere und stabilere Benutzererfahrung bieten, sind die OSS Redis-Versionen 2.6.13, 2.8.6 und 2.8.19 nicht mehr in der Management Console aufgeführt. ElastiCache Weitere Optionen und mehr Informationen finden Sie unter <a href="#">Unterstützte Engines und Versionen</a> .	15. Dezember 2015

Änderung	Beschreibung	Änderungsdatum
Support für Redis OSS 2.8.23.	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.23 hinzu, wobei Verbesserungen seit Redis 2.8.22 hinzugefügt wurden. OSS Die Verbesserungen umfassen Fehlerkorrekturen sowie Support für den neuen Parameter <code>close-on-slave-write</code> der in aktiviertem Zustand die Verbindung zu Clients trennt, die versuchen, in ein schreibgeschütztes Replikat zu schreiben. Weitere Informationen finden Sie unter <a href="#">ElastiCache (RedisOSS) Version 2.8.23 (erweitert)</a> .	13. November 2015

Änderung	Beschreibung	Änderungsdatum
Support für Redis OSS 2.8.22.	<p>ElastiCache fügt Unterstützung für Redis OSS Version 2.8.22 mit ElastiCache zusätzlichen Erweiterungen und Verbesserungen seit Version 2.8.21 hinzu. Zu den Verbesserungen gehören:</p> <ul style="list-style-type: none"><li data-bbox="402 457 1122 709">• Implementierung eines unvergabelten Speicherprozesses, der eine erfolgreiche Speicherung ermöglicht, wenn aufgrund wenig verfügbarem Arbeitsspeicher eine vergabelte Speicherung fehlschlagen würde.</li><li data-bbox="402 737 1024 846">• Zusätzliche Metriken — und. CloudWatch SaveInProgressReplicationBytes</li><li data-bbox="402 873 1089 1031">• Um Teilsynchronisationen zu ermöglichen, gilt der OSS Redis-Parameter <code>repl-backlog-size</code> jetzt für alle Cluster.</li></ul> <p>Eine komplette Liste der Änderungen und weitere Informationen finden Sie unter <a href="#">ElastiCache (RedisOSS) Version 2.8.22 (erweitert)</a>.</p> <p>Diese Dokumentationsversion beinhaltet eine Neuorganisation der Dokumentation und die Entfernung der Dokumentation zur ElastiCache Befehlszeilenschnittstelle (CLI). Informationen zur Verwendung der Befehlszeile finden Sie in der <a href="#">AWS Befehlszeile</a> für ElastiCache.</p>	28. September 2015

Änderung	Beschreibung	Änderungsdatum
Unterstützung für Memcached 1.4.28	ElastiCache fügt Unterstützung für Memcached Version 1.4.24 und Memcached-Verbesserungen seit Version 1.4.14 hinzu. Diese Version bietet Unterstützung für die Cache-Verwaltung (LRU), die zuletzt als Hintergrundaufgabe verwendet wurde, die Wahl von Jenkins oder Murmur3 als Hash-Algorithmus, neue Befehle und verschiedene Bugfixes. Weitere Informationen finden Sie unter den <a href="#">Memcached-Versionshinweisen</a> .	27. August 2015
Support für Memcached Auto Discovery mit 5.6 PHP	Diese Version von Amazon ElastiCache bietet Unterstützung für den Memcached Auto Discovery -Client für PHP Version 5.6. Weitere Informationen finden Sie unter <a href="#">Kompilieren des Quellcodes für den ElastiCache Cluster-Client für PHP</a> .	29. Juli 2015
Support für Redis 2.8.21 OSS	ElastiCache fügt Unterstützung für Redis OSS Version 2.8.21 und Redis-Verbesserungen seit Version 2.8.19 hinzu. Diese OSS Redis-Version enthält mehrere Bugfixes. Weitere Informationen finden Sie in den <a href="#">Versionshinweisen zu Redis OSS 2.8</a> .	29. Juli 2015
Neues Thema: Zugriff ElastiCache von außen AWS	Es wurde ein neues Thema zum Zugriff auf ElastiCache Ressourcen von außen hinzugefügt AWS. Weitere Informationen finden Sie unter <a href="#">Zugriff ElastiCache von außen AWS</a> .	9. Juli 2015

Änderung	Beschreibung	Änderungsdatum
Nachrichten über Knotenersatz wurden hinzugefügt	<p>ElastiCache fügt drei Meldungen hinzu, die sich auf den geplanten Knotenaustausch beziehen: <code>ElastiCache: NodeReplacementScheduled</code>, <code>ElastiCache: NodeReplacementRescheduled</code>, und <code>ElastiCache: NodeReplacementCanceled</code>.</p> <p>Weitere Informationen und Maßnahmen, die Sie ergreifen können, wenn der Austausch eines Knotens geplant ist, finden Sie unter <a href="#">ElastiCache. Event-Benachrichtigungen und Amazon SNS</a></p>	11. Juni 2015

Änderung	Beschreibung	Änderungsdatum
Support für Redis OSS v. 2.8.19.	<p>ElastiCache fügt Unterstützung für Redis OSS Version 2.8.19 und Redis-Verbesserungen seit Version 2.8.6 hinzu. OSS Diese Unterstützung enthält Unterstützung für:</p> <ul style="list-style-type: none"> <li>• Die HyperLogLog Datenstruktur mit den OSS Redis-Befehlen, und. PFADD PFCOUNT PFMERGE</li> <li>• Lexikographische Bereichsabfragen mit den neuen BefehlenZRANGEBYLEX, undZLEXCOUNT. ZREMRANGEBYLEX</li> <li>• Es wurden eine Reihe von Bugfixes eingeführt, insbesondere wurde verhindert, dass ein primärer Knoten veraltete Daten an Replikatknoten sendet, indem der primäre Knoten ausfällt, SYNC wenn ein untergeordneter Prozess zum Speichern im Hintergrund (bgsave) unerwartet beendet wird.</li> </ul> <p>Weitere Informationen dazu finden Sie unter <a href="#">Redis OSS neue</a> Datenstruktur: die. HyperLogLog HyperLogLog</p> <p>Weitere Informationen zu PFADDPFCOUNT, und PFMERGE finden Sie in der <a href="#">OSSRedis-Dokumentation</a> und klicken Sie auf. HyperLogLog</p>	11. März 2015
Unterstützung von Kostenzuordnungs-Tags	ElastiCache fügt Unterstützung für Kostenzuordnungs-Tags hinzu. Weitere Informationen finden Sie unter <a href="#">Überwachung von Kosten mit Kostenzuordnungs-Tags</a> .	9. Februar 2015

Änderung	Beschreibung	Änderungsdatum
Support für die AWS GovCloud Region (US-West)	ElastiCache fügt Unterstützung für die Region AWS GovCloud (US-West) (us-gov-west-1) hinzu.	29. Januar 2015
Support für die Region Europa (Frankfurt)	ElastiCache fügt Unterstützung für die Region Europa (Frankfurt) (eu-central-1) hinzu.	19. Januar 2015
Multi-AZ-Unterstützung für Redis-Replikationsgruppen OSS	ElastiCache fügt Unterstützung für Multi-AZ vom primären Knoten zu einer Read Replica in einer Redis-Replikationsgruppe hinzu. OSS ElastiCache überwacht den Zustand der Replikationsgruppe. Wenn das Primärreplikat ausfällt, ElastiCache wird ein Replikat automatisch zum Primärreplikat heraufgestuft und dann das Replikat ersetzt. Weitere Informationen finden Sie unter <a href="#">Minimierung von Ausfallzeiten durch die Verwendung ElastiCache von Multi-AZ mit Valkey und Redis OSS</a> .	24. Oktober 2014
AWS CloudTrail Protokollierung von Anrufen wird unterstützt API	ElastiCache fügt Unterstützung für die Verwendung AWS CloudTrail zum Protokollieren aller ElastiCache API Anrufe hinzu. Weitere Informationen finden Sie unter <a href="#">ElastiCache API Amazon-Anrufe protokollieren mit AWS CloudTrail</a> .	15. September 2014
Es werden neue Instance-Größen unterstützt	ElastiCache fügt Unterstützung für zusätzliche General Purpose (T2) -Instances hinzu. Weitere Informationen finden Sie unter <a href="#">Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen</a> .	11. September 2014

Änderung	Beschreibung	Änderungsdatum
Flexibler Knotenersatz für Memcached wird unterstützt	ElastiCache fügt Unterstützung für die Erstellung von Memcached-Knoten in mehreren Availability Zones hinzu.	23. Juli 2014
Es werden neue Instance-Größen unterstützt	ElastiCache fügt Unterstützung für zusätzliche General Purpose (M3) -Instances und Memory Optimized (R3) -Instances hinzu. Weitere Informationen finden Sie unter <a href="#">Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen</a> .	1. Juli 2014
PHPauto Erkennung	Unterstützung für PHP Version 5.5 Auto Discovery hinzugefügt.	13. Mai 2014
Backup und Wiederherstellung für Redis-Cluster OSS	In dieser Version ElastiCache können Kunden Snapshots ihrer OSS Redis-Cluster erstellen und mithilfe dieser Snapshots neue Cluster erstellen. Ein Backup ist eine Kopie des Clusters zu einem bestimmten Zeitpunkt und besteht aus Cluster-Metadaten und allen Daten im Redis-Cache. OSS Sicherungen werden in Amazon S3 gespeichert. Kunden können die Daten von einem Snapshot jederzeit in einem neuen Cluster wiederherstellen. Weitere Informationen finden Sie unter <a href="#">Snapshot und Wiederherstellung</a> .	24. April 2014

Änderung	Beschreibung	Änderungsdatum
Redis 2.8.6 OSS	ElastiCache unterstützt Redis OSS 2.8.6 zusätzlich zu Redis 2.6.13. OSS Mit Redis OSS 2.8.6 können Kunden die Stabilität und Fehlertoleranz von Read Replicas verbessern, indem sie die teilweise Resynchronisierung unterstützen und eine benutzerdefinierte Mindestanzahl von Read Replicas festlegen, die jederzeit verfügbar sein müssen. Redis OSS 2.8.6 bietet auch volle Unterstützung für publish-and-subscribe, sodass Clients über Ereignisse auf dem Server informiert werden können.	13. März 2014
Redis-Cache-Engine OSS	<p>ElastiCache bietet zusätzlich zu OSS Memcached die Redis-Cache-Engine-Software an. Kunden, die derzeit Redis verwenden, OSS können einen neuen OSS Redis-Cache-Cluster mit ihren vorhandenen Daten aus einer ElastiCache OSS Redis-Snapshot-Datei „seeding“ erstellen und so die Migration zu einer verwalteten Umgebung vereinfachen. ElastiCache</p> <p>Zur Unterstützung der OSS Redis-Replikationsfunktionen unterstützt der ElastiCache API jetzt Replikationsgruppen. Kunden können eine Replikationsgruppe mit einem primären OSS Redis-Cache-Knoten erstellen und einen oder mehrere Read Replica-Knoten hinzufügen, die automatisch mit den Cache-Daten im primären Knoten synchronisiert bleiben. Leselast-Anwendungen können in ein Lesereplikat ausgelagert werden, wodurch die Arbeitslast auf dem Primärknoten reduziert wird. Lesereplikate können auch vor Datenverlust im Fall eines Ausfalls des primären Cache-Knotens schützen.</p>	3. September 2013

Änderung	Beschreibung	Änderungsdatum
Support für die standardmäßige Amazon Virtual Private Cloud (VPC)	In dieser Version ElastiCache ist es vollständig in Amazon Virtual Private Cloud (VPC) integriert. Für Neukunden werden Cache-Cluster VPC standardmäßig in einem Amazon erstellt. Weitere Informationen finden Sie unter <a href="#">Amazon VPCs und ElastiCache Sicherheit</a> .	8. Januar 2013
PHP-Unterstützung für die auto Erkennung von Cache-Knoten	Die ursprüngliche Version von Auto Discovery für Cache-Knoten bot Unterstützung für Java-Programme. In dieser Version wird ElastiCache die auto Erkennung von Cache-Knoten unterstützt PHP.	2. Januar 2013
Support für Amazon Virtual Private Cloud (VPC)	In dieser Version können ElastiCache Cluster in Amazon Virtual Private Cloud (VPC) gestartet werden. Standardmäßig werden Cache-Cluster für neue Kunden VPC automatisch in Amazon erstellt. Bestandskunden können in ihrem eigenen Tempo zu Amazon VPC migrieren. Weitere Informationen finden Sie unter <a href="#">Amazon VPCs und ElastiCache Sicherheit</a> .	20. Dezember 2012

Änderung	Beschreibung	Änderungsdatum
Auto Discovery für Cache-Knoten und neue Cache-Engine-Version	<p>ElastiCache bietet die auto von Cacheknoten, d. h. die Fähigkeit von Client-Programmen, automatisch alle Cache-Knoten in einem Cluster zu ermitteln und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten.</p> <p>Diese Version bietet außerdem eine neue Cache-Engine-Version, Memcached 1.4.14. Diese neue Cache-Engine bietet erweiterte Slab-Ausgleichsfunktionen, entscheidende Leistungs- und Skalierungsverbesserungen sowie mehrere Fehlerkorrekturen. Es gibt verschiedene neue Cache-Parameter, die konfiguriert werden können. Weitere Informationen finden Sie unter <a href="#">Konfiguration von Motorparametern mithilfe von ElastiCache Parametergruppen</a>.</p>	28. November 2012
Neue Cache-Knotentypen	Diese Version stellt vier neue Cache-Knotentypen bereit.	13. November 2012
Reservierte Cache-Knoten	Diese Version bietet nun Unterstützung für reservierte Cache-Knoten.	5. April 2012
Neues Handbuch	Dies ist die erste Version des ElastiCache Amazon-Benutzerhandbuchs.	22. August 2011

# AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.