



Leitfaden

Amazon ElastiCache für Redis



API-Version 2015-02-02

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

Amazon ElastiCache für Redis: 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 für Redis? | 1 |
| Serverless-Caching | 1 |
| Selbst entworfene Cluster | 2 |
| Zugehörige Services | 2 |
| Funktionsweise | 3 |
| Cache- und Caching-Engines | 3 |
| Auswählen der Bereitstellung | 8 |
| Vergleichen von Features | 10 |
| ElastiCache-Ressourcen | 15 |
| AWS-Regionen und Availability Zones | 17 |
| Anwendungsfälle | 18 |
| In-Memory-Datastore | 19 |
| Gaming Leaderboards (Sortierte Sätze von Redis) | 20 |
| Messaging (Redis Pub/Sub) | 22 |
| Empfehlungsdaten (Redis-Hashes) | 25 |
| Andere Verwendung von Redis | 26 |
| Kundenstimmen zu ElastiCache | 26 |
| Erste Schritte mit ElastiCache for Redis | 27 |
| Einrichtung | 27 |
| Melde dich an für ein AWS-Konto | 27 |
| Erstellen Sie einen Benutzer mit Administratorzugriff | 28 |
| Erteilen programmgesteuerten Zugriffs | 29 |
| Berechtigungen einrichten | 31 |
| Einrichten von EC2 | 32 |
| Gewähren des Netzwerkzugriffs | 33 |
| Einrichten von redis-cli | 34 |
| Erstellen eines Caches | 34 |
| Lesen und Schreiben von Daten | 36 |
| Bereinigen | 38 |
| Nächste Schritte | 39 |
| Erste Schritte mit ElastiCache und AWS-SDKs | 39 |
| Python und ElastiCache | 39 |
| Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC | 57 |

| | |
|--|-----|
| Schritt 1: Einen serverlosen Cache erstellen | 58 |
| Schritt 2: Erstellen einer Lambda-Funktion | 61 |
| Schritt 3: Testen der Lambda -Funktion | 65 |
| Schritt 4: Aufräumen (optional) | 65 |
| Entwerfen eines eigenen ElastiCache-Clusters | 68 |
| Komponenten und Funktionen | 68 |
| Knoten | 69 |
| ElastiCache für Redis-Shards | 70 |
| ElastiCache für Redis-Cluster | 70 |
| ElastiCache für die Redis-Replikation | 72 |
| AWS Regionen und Verfügbarkeitszonen | 74 |
| ElastiCache für Redis-Endpunkte | 75 |
| Parametergruppen | 76 |
| ElastiCache für die Redis-Sicherheit | 76 |
| Subnetzgruppen | 76 |
| ElastiCache für Redis-Backups | 77 |
| Ereignisse | 77 |
| ElastiCache-for-Redis-Terminologie | 78 |
| Entwerfen eines eigenen Clusters | 81 |
| Einrichtung | 81 |
| Schritt 1: Erstellung einer Subnetzgruppe | 81 |
| Schritt 2: Erstellen eines Clusters | 84 |
| Schritt 3: Zugriff auf den Cluster autorisieren | 92 |
| Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten | 95 |
| Schritt 5: Löschen eines Clusters | 103 |
| Tutorials und Videos | 105 |
| Wie geht es weiter? | 111 |
| Knoten verwalten | 112 |
| ElastiCache Knotenstatus anzeigen | 113 |
| Redis-Knoten und -Shards | 119 |
| Mit Knoten verbinden | 121 |
| Unterstützte Knotentypen | 124 |
| Neustart von Knoten (nur mit deaktiviertem Cluster-Modus) | 135 |
| Ersetzen von Knoten | 137 |
| Reservierte Knoten | 144 |
| Migration von Knoten der vorherigen Generation | 156 |

| | |
|--|-----|
| Verwalten von Clustern | 159 |
| Auswählen eines Netzwerktyps | 161 |
| Daten-Tiering | 165 |
| Vorbereitung eines Clusters | 172 |
| Erstellen eines Clusters | 180 |
| Anzeigen der Details eines Clusters | 191 |
| Modifizieren eines Clusters | 204 |
| Knoten zu einem Cluster hinzufügen | 210 |
| Entfernen von Knoten aus einem Cluster | 218 |
| Abbrechen ausstehender Vorgänge zum Hinzufügen oder Löschen von Knoten | 226 |
| Löschen eines Clusters | 227 |
| Zugriff auf Ihren Cluster oder die Replikationsgruppe | 230 |
| Verbindungsendpunkte ermitteln | 237 |
| Shards | 248 |
| Vergleich der selbst entworfenen Caches von Memcached und Redis | 253 |
| Online-Migration zu ElastiCache | 258 |
| Übersicht | 259 |
| Schritte zur Migration | 260 |
| Vorbereiten Ihrer Redis-Quell- und -Zielknoten für die Migration | 260 |
| Testen der Datenmigration | 262 |
| Migration starten | 262 |
| Verifizieren des Fortschritts der Datenmigration | 264 |
| Datenmigration abschließen | 264 |
| Durchführen einer Online-Datenmigration mithilfe der Konsole | 265 |
| Auswahl von Regionen und Availability Zones | 267 |
| Lokalisieren Ihrer Knoten | 269 |
| Unterstützte Regionen und Endpunkte | 269 |
| Local Zones verwenden | 274 |
| Verwenden von Outposts | 276 |
| Arbeiten mit ElastiCache | 280 |
| Snapshot und Wiederherstellung | 280 |
| Beschränkungen | 281 |
| Auswirkungen von Backups selbst entworfener Cluster auf die Leistung | 282 |
| Planen automatischer Backups | 284 |
| Erstellen manueller Backups | 285 |
| Erstellen einer endgültigen Sicherung | 291 |

| | |
|---|-----|
| Beschreiben von Sicherungen | 294 |
| Kopieren eines Backups | 296 |
| Exportieren einer Sicherung | 299 |
| Wiederherstellen aus einem Backup | 307 |
| Löschen einer Sicherung | 309 |
| Markieren von Sicherungen | 311 |
| Seeding eines selbst entworfenen Clusters mit einem Backup | 312 |
| Engine-Versionen und Upgrades | 322 |
| Engine-Versionen und Upgrades | 323 |
| Unterstützte Redis-Versionen | 328 |
| Zeitplan für das Lebensende der Redis-Versionen | 342 |
| So führen Sie ein Upgrade von Engine-Versionen aus | 326 |
| Beheben blockierter Engine-Upgrades | 326 |
| Wesentliche Versionsverhalten und Kompatibilitätsunterschiede | 345 |
| Bewährte Methoden und Caching-Strategien | 349 |
| Arbeiten mit Redis | 350 |
| Bewährte Methoden für Redis-Clients | 390 |
| Verwalten von reserviertem Speicher | 418 |
| Bewährte Methoden bei der Arbeit mit selbst entworfenen Clustern | 425 |
| Bewährte Methoden für Redis | 431 |
| Caching-Strategien | 433 |
| Verwalten Ihrer selbst entworfenen Cluster | 438 |
| Auto Scaling ElastiCache für Redis-Cluster | 439 |
| Ändern des Clustermodus | 487 |
| AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher | 491 |
| Hohe Verfügbarkeit mit Replikationsgruppen | 519 |
| Verwaltung der Wartung | 609 |
| Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen | 612 |
| Skalierung ElastiCache für Redis | 715 |
| Serverlose Skalierung ElastiCache | 715 |
| Festlegen von Skalierungslimits zur Kostenkontrolle | 715 |
| Vorskalierung mit Serverless ElastiCache | 715 |
| Skalierungsgrenzen mithilfe der Konsole festlegen und AWS CLI | 717 |
| Skalierung ElastiCache für selbst entwickelte Redis-Cluster | 718 |
| Erste Schritte mit JSON in ElastiCache for Redis | 792 |
| Übersicht über Redis-JSON-Datentypen | 793 |

| | |
|---|------|
| JSON-Befehle | 805 |
| Markieren Ihrer ElastiCache-Ressourcen | 847 |
| Überwachung von Kosten mit Tags | 859 |
| Verwalten von Tags mit dem AWS CLI | 861 |
| Verwalten von Tags mithilfe der ElastiCache-API | 865 |
| Amazon ElastiCache Well-Architected Lens | 867 |
| Säule „Operational Excellence“ | 868 |
| Säule der Sicherheit | 877 |
| Säule der Zuverlässigkeit | 884 |
| Säule der Leistungseffizienz | 890 |
| Säule der Kostenoptimierung | 902 |
| Fehlerbehebung | 909 |
| Verbindungsprobleme | 909 |
| Redis-Clientfehler | 910 |
| Fehlerbehebung bei hoher Latenz in ElastiCache Serverless | 911 |
| Fehlerbehebung bei Drosselungsproblemen in ElastiCache Serverless | 913 |
| Verwandte Themen | 914 |
| Zusätzliche Schritte zur Fehlerbehebung | 914 |
| Sicherheitsgruppen | 914 |
| Netzwerk-ACLs | 915 |
| Routing-Tabellen | 917 |
| DNS-Auflösung | 917 |
| Identifizieren von Problemen mit serverseitiger Diagnose | 917 |
| Netzwerkverbindbarkeit | 924 |
| Netzwerkbezogene Grenzen | 926 |
| CPU-Verwendung | 928 |
| Verbindungen, die von der Serverseite beendet werden | 931 |
| Clientseitige Fehlerbehebung für Amazon EC2 Instances | 932 |
| Aufschlüsselung der Zeit, die zum Abschließen einer einzelnen Anfrage benötigt wird | 933 |
| Sicherheit | 937 |
| Datenschutz | 938 |
| Datensicherheit in Amazon ElastiCache | 939 |
| Richtlinie für den Datenverkehr zwischen Netzwerken | 1015 |
| Amazon VPCs und ElastiCache Sicherheit | 1015 |
| Amazon-ElastiCache-API und Schnittstellen-VPC-Endpunkte (AWS PrivateLink) | 1042 |
| Subnetze und Subnetzgruppen | 1045 |

| | |
|--|------|
| Identitäts- und Zugriffsverwaltung | 1054 |
| Zielgruppe | 1054 |
| Authentifizierung mit Identitäten | 1055 |
| Verwalten des Zugriffs mit Richtlinien | 1059 |
| So ElastiCache arbeitet Amazon mit IAM | 1062 |
| Beispiele für identitätsbasierte Richtlinien | 1070 |
| Fehlerbehebung | 1073 |
| Zugriffskontrolle | 1075 |
| Übersicht über die Verwaltung von Zugriffsberechtigungen | 1076 |
| Compliance-Validierung | 1120 |
| Weitere Informationen | 1122 |
| Ausfallsicherheit | 1122 |
| Minimieren von Ausfällen | 1123 |
| Sicherheit der Infrastruktur | 1126 |
| Service-Updates | 1127 |
| Verwaltung von Service-Updates | 1127 |
| Sicherheitslücken wurden behoben | 1133 |
| Protokollierung und Überwachung | 1135 |
| Serverless-Metriken und -Ereignisse | 1135 |
| Serverless-Metriken | 1135 |
| Serverless-Ereignisse | 1145 |
| Metriken und Ereignisse für selbst entworfene Cluster | 1162 |
| Metriken für selbst entworfene Cluster | 1162 |
| Ereignisse für selbst entworfene Cluster | 1162 |
| Protokollzustellung | 1171 |
| Überwachung der Nutzung | 1184 |
| Amazon-SNS-Ereignisüberwachung | 1214 |
| Protokollieren von Amazon-ElastiCache-API-Aufrufen mithilfe von AWS CloudTrail | 1232 |
| Informationen zu Amazon ElastiCache in CloudTrail | 1232 |
| Grundlegendes zu den Amazon-ElastiCache-Einträgen in der Protokolldatei | 1233 |
| Kontingente | 1238 |
| Referenz | 1240 |
| Verwenden der ElastiCache-API | 1240 |
| Verwenden der Abfrage-API | 1240 |
| Verfügbare Bibliotheken | 1244 |
| Fehlerbehebung bei Anwendungen | 1244 |

| | |
|---|---------|
| Einrichten der AWS CLI für ElastiCache | 1246 |
| Voraussetzungen | 1246 |
| Aufrufen der Befehlszeilen-Tools | 1248 |
| Einrichten der Tools | 1248 |
| Bereitstellung von Anmeldeinformationen für die Tools | 1249 |
| Umgebungsvariablen | 1251 |
| Fehlermeldungen | 1251 |
| Benachrichtigungen | 1253 |
| Allgemeine Benachrichtigungen von ElastiCache | 1253 |
| Spezifische ElastiCache-for-Redis-Benachrichtigungen | 1254 |
| Dokumentationsverlauf ElastiCache für Redis | 1255 |
| AWS Glossar | 1291 |
| | mccxcii |

Was ist Amazon ElastiCache für Redis?

Willkommen beim Amazon ElastiCache -for-Redis-Benutzerhandbuch. Amazon ElastiCache ist ein Webservice, der das Einrichten, Verwalten und Skalieren eines verteilten In-Memory-Datenspeichers oder einer Cache-Umgebung in der Cloud vereinfacht. 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 sowohl mit den Redis- als auch mit den Memcached-Engines. Verwenden Sie das Handbuch für die relevante Engine. Wenn Sie sich nicht sicher sind, welche Engine Sie verwenden möchten, beachten Sie [Vergleich der selbst entworfenen Caches von Memcached und Redis](#) in diesem Handbuch.

Serverless-Caching

ElastiCache for Redis bietet Serverless-Caching, was das Hinzufügen und Betreiben eines Redis-basierten Cache für Ihre Anwendung vereinfacht. ElastiCache for Redis Serverless ermöglicht es Ihnen, einen hochverfügbaren Cache in weniger als einer Minute zu erstellen, und macht es überflüssig, Instances bereitzustellen oder Knoten oder Cluster zu konfigurieren. Entwickler können einen Serverless-Cache erstellen, indem sie den Cache-Namen mithilfe der ElastiCache Konsole, des SDK oder der CLI angeben.

ElastiCache für Redis Serverless macht auch die Planung und Verwaltung der Caching-Kapazität überflüssig. ElastiCache für Redis überwacht ständig den Arbeitsspeicher des Cache. Datenverarbeitung, und Netzwerkbandbreite, die von Ihrer Anwendung verwendet wird, und werden entsprechend den Anforderungen Ihrer Anwendung skaliert. ElastiCache for Redis bietet ein einfaches Endpunkterlebnis für Entwickler. durch Abstrahieren der zugrunde liegenden Cache-Infrastruktur und des Clusterdesigns. ElastiCache für Redis verwaltet die Hardwarebereitstellung. - Überwachung, Knotenaustausche, und Software-Patching automatisch und transparent, sodass Sie sich auf die Anwendungsentwicklung konzentrieren können, statt den Cache zu betreiben.

ElastiCache für Redis Serverless ist mit Redis 7.1 und höher kompatibel.

Entwerfen Ihres eigenen ElastiCache für Redis-Cluster

Wenn Sie eine differenzierte Kontrolle über Ihren ElastiCache für Redis-Cluster benötigen, können Sie Ihren eigenen Redis-Cluster mit entworfen ElastiCache. ElastiCache ermöglicht es Ihnen, Ihren Cluster zu entwerfen, indem Sie den Knotentyp, die Anzahl der Knoten und die Knotenplatzierung über AWS Availability Zones für Ihren Cluster hinweg auswählen. Da ein vollständig verwalteter Service ElastiCache ist, werden Hardwarebereitstellung, Überwachung, Knotenaustausch und Software-Patching für Ihren Cluster automatisch verwaltet.

Das Entwerfen Ihres eigenen ElastiCache für Redis-Cluster 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 sich auch dafür entscheiden, Redis im Cluster-Modus auszuführen, um horizontale Skalierung zu ermöglichen, oder ohne Cluster-Modus, wenn nur vertikal skaliert werden soll. 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 Redis-Cluster angewendet werden sollen.

Wenn Sie Ihren eigenen ElastiCache für Redis-Cluster entwerfen, können Sie Redis 3.0 und höher ausführen.

Zugehörige Services

[Amazon MemoryDB für Redis](#)

Ziehen Sie bei der Entscheidung, ob Sie ElastiCache für Redis oder Amazon MemoryDB für Redis verwenden möchten, die folgenden Vergleiche in Betracht:

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

- Amazon MemoryDB für Redis ist eine dauerhafte In-Memory-Datenbank für Workloads, die eine ultraschnelle Primärdatenbank erfordern. 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 mithilfe von 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 RDS](#)

ElastiCache für Redis 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 mit ElastiCache eine hohe Skalierbarkeit, schnelle Leistung und geringere Datenspeicherkosten erzielen, anstatt Ihre zugrunde liegende Datenbank zu skalieren.

Funktionsweise

Hier finden Sie einen Überblick über die Hauptkomponenten einer ElastiCache für Redis-Bereitstellung.

Cache- und Caching-Engines

Ein Cache ist ein In-Memory-Datenspeicher, mit dem Sie zwischengespeicherte Daten speichern können. In der Regel speichert Ihre Anwendung häufig aufgerufene Daten in einem Cache, um die Antwortzeiten zu optimieren. ElastiCache für Redis bietet zwei Bereitstellungsoptionen: Serverless und selbst entwickelte Cluster. Siehe [Auswählen aus den Bereitstellungsoptionen](#)

Note

Amazon ElastiCache arbeitet sowohl mit den Redis- als auch mit den Memcached-Engines. Verwenden Sie das Handbuch für die relevante Engine. Wenn Sie sich nicht sicher sind, welche Engine Sie verwenden möchten, beachten Sie [Vergleich der selbst entworfenen Caches von Memcached und Redis](#) in diesem Handbuch.

Themen

- [Funktionsweise von ElastiCache für Redis](#)
- [Preisdimensionen](#)
- [ElastiCache für Redis-Backups](#)

Funktionsweise von ElastiCache für Redis

ElastiCache für Redis Serverless

ElastiCache for Redis Serverless ermöglicht es Ihnen, einen Cache zu erstellen, ohne sich um Kapazitätsplanung, Hardwaremanagement oder Clusterdesign kümmern zu müssen. Sie geben einfach einen Namen für Ihren Cache ein und erhalten einen einzigen Endpunkt, den Sie in Ihrem Redis-Client konfigurieren können, um den Zugriff auf Ihren Cache zu ermöglichen.

Note

ElastiCache for Redis Serverless führt Redis im Cluster-Modus aus und ist nur mit Redis-Clients kompatibel, die sowohl TLS als auch das Redis-Cluster-Protokoll unterstützen.

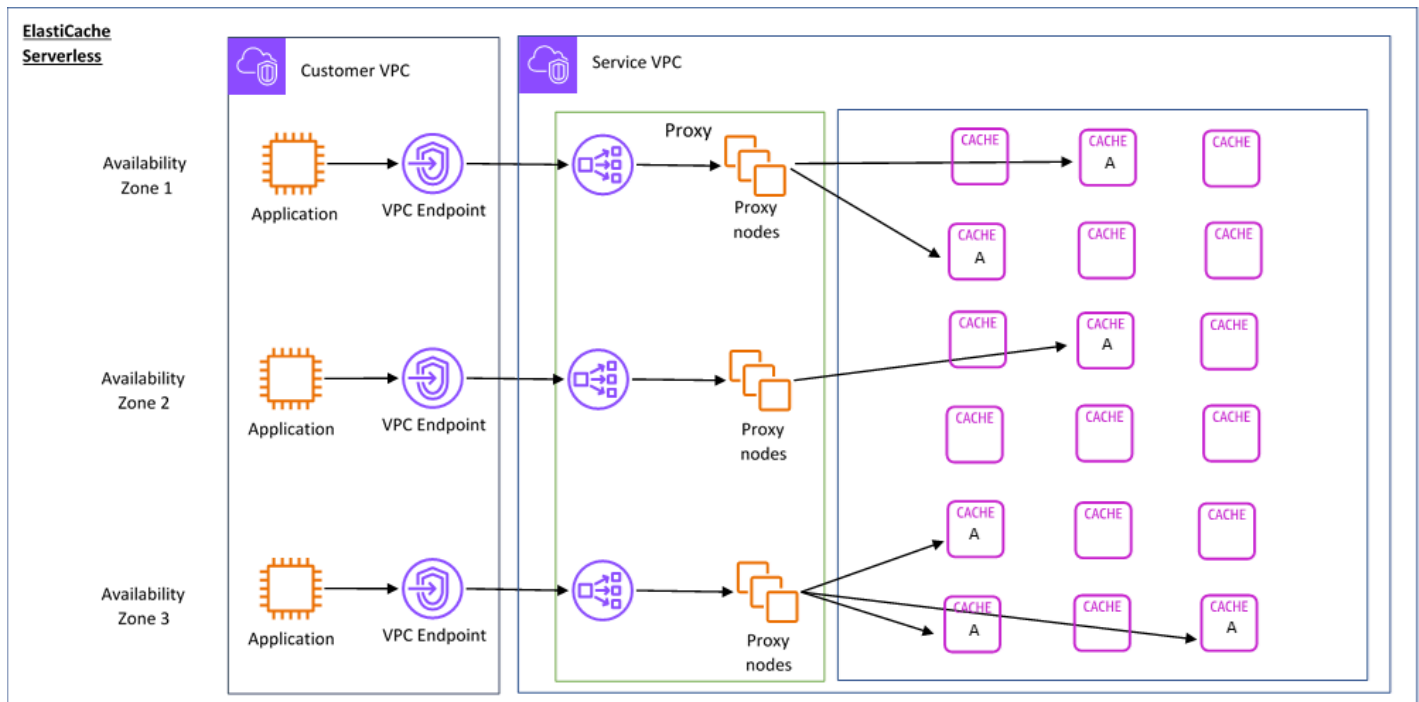
Wichtigste Vorteile

- **Keine Kapazitätsplanung:** ElastiCache Serverless macht die Planung der Kapazität überflüssig. ElastiCache Serverless überwacht kontinuierlich die Speicher-, Rechen- und Netzwerkbandbreitenauslastung Ihres Cache 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 gespeicherten Daten und die Datenverarbeitung, die von Ihrem Workload im Cache genutzt werden. Siehe [Preisdimensionen](#).
- **Hohe Verfügbarkeit:** 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 ein SLA mit einer Verfügbarkeit von 99,99 % für jeden Cache.
- **Automatische Software-Upgrades:** ElastiCache Serverless aktualisiert Ihren Cache automatisch auf die neueste Neben- und Patch-Softwareversion, ohne dass sich dies auf die Verfügbarkeit Ihrer

Anwendung auswirkt. Wenn eine neue Redis-Hauptversion verfügbar ist, ElastiCache sendet Ihnen 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 veranschaulicht, wie ElastiCache Serverless funktioniert.



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

Mit ElastiCache Serverless erhalten Sie einen einzelnen 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 Proxy-Ebene. 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. Auf diese Weise können neue Cache-Knoten ElastiCache intelligent geladen, aufskaliert und hinzugefügt, Cache-Knoten ersetzt werden, wenn sie ausfallen, und Software auf den Cache-Knoten aktualisiert werden, ohne dass sich dies auf Ihre Anwendung auswirkt oder Verbindungen zurücksetzen muss.

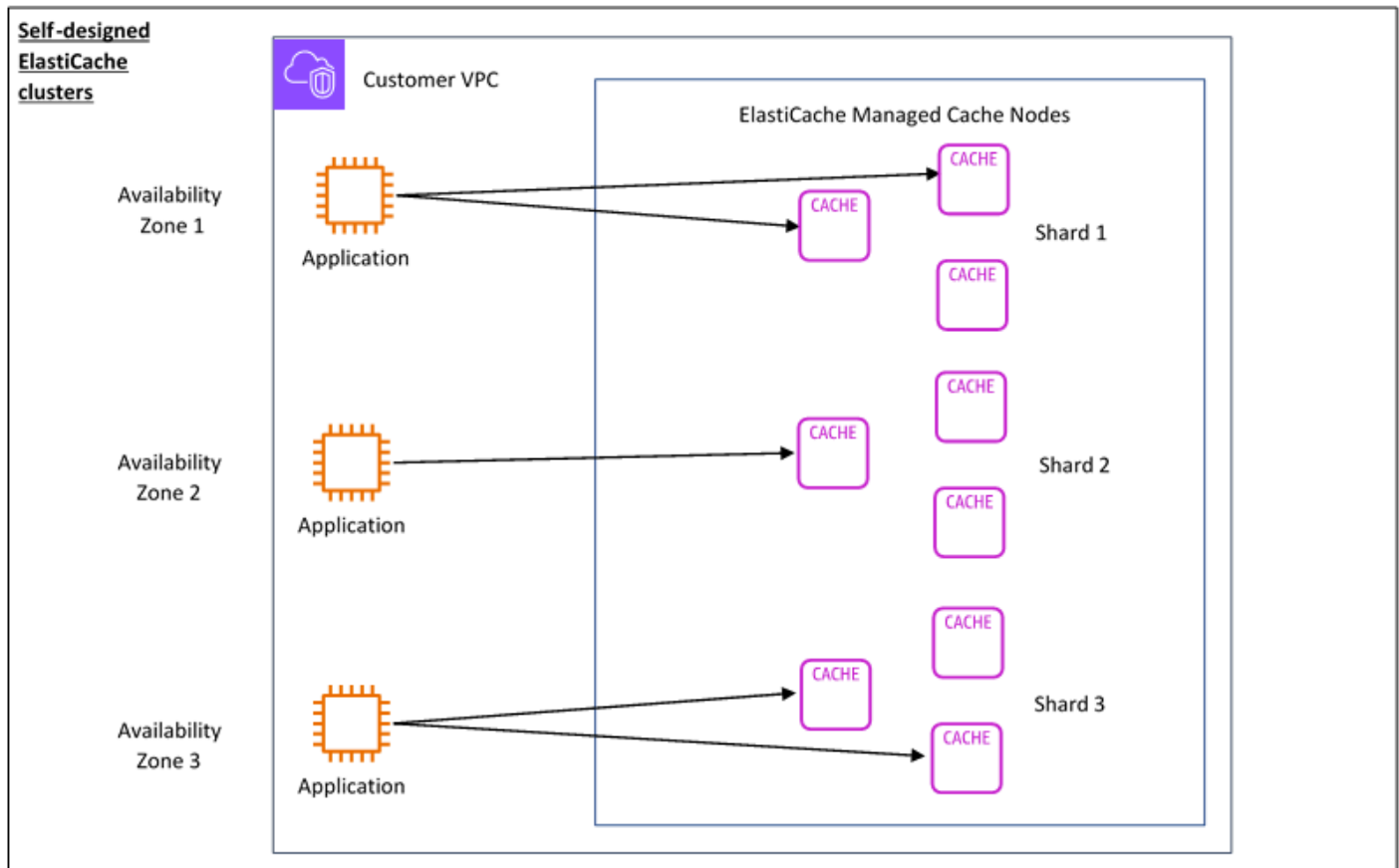
Selbstentwerfene ElastiCache Cluster

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 Redis im Cluster-Modus betreiben möchten, indem Sie einen Cluster mit mehreren Shards erstellen, oder im Nicht-Cluster-Modus mit einem einzelnen Shard.

Wichtigste Vorteile

- Entwerfen Sie Ihren eigenen Cluster: Mit können Sie Ihren eigenen Cluster entwerfen und auswählen 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 in mehreren AZs entwerfen, 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. Sie können beispielsweise [Redis-spezifische Parameter](#) verwenden, 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 basierend auf einem Zeitplan oder die Skalierung basierend auf Metriken wie CPU- und Speicherauslastung im Cache zu konfigurieren.

Das folgende Diagramm veranschaulicht, wie ElastiCache selbstentwickelte Cluster funktionieren.



Preisdimensionen

Sie können ElastiCache in zwei Bereitstellungsoptionen bereitstellen. Bei der Bereitstellung von ElastiCache Serverless zahlen Sie für die Nutzung von Daten, die in GB-Stunden gespeichert sind, und für die Datenverarbeitung in ElastiCache Processing Units (ECPU). Wenn Sie Ihre eigenen ElastiCache für Redis-Cluster entwerfen möchten, zahlen Sie pro Stunde der Cache-Knotennutzung. 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, erfasst mehrere Male pro Minute und berechnet einen stündlichen Durchschnitt, um die Datenspeichernutzung des Cache in GB-Stunden zu bestimmen. Jeder ElastiCache Serverless-Cache wird für mindestens 1 GB gespeicherter Daten gemessen.

ElastiCache Verarbeitungseinheiten (ECPUs)

Sie zahlen für die Redis-Anforderungen, die Ihre Anwendung auf ElastiCache Serverless in ElastiCache Processing Units (ECPUs) ausführt, einer Einheit, die sowohl vCPU-Zeit als auch übertragene Daten enthält.

- Einfache Lese- und Schreibvorgänge erfordern 1 ECPU für jedes Kilobyte (KB) übertragener Daten. Beispielsweise verbraucht ein GET-Befehl, der bis zu 1 KB Daten überträgt, 1 ECPU. Eine SET-Anfrage, die 3,2 KB Daten überträgt, verbraucht 3,2 ECPUs.
- Befehle, die zusätzliche vCPU-Zeit benötigen, verbrauchen proportional mehr ECPUs. Wenn Ihre Anwendung beispielsweise den Redis-Befehl [HMGET](#) verwendet und das Dreifache der vCPU-Zeit als ein einfacher SET/GET-Befehl benötigt, verbraucht sie 3 ECPUs.
- Befehle, die mehr vCPU-Zeit benötigen und mehr Daten übertragen, verbrauchen ECPUs, die auf der höheren der beiden Dimensionen basieren. Wenn Ihre Anwendung beispielsweise den Befehl HMGET verwendet, das Dreifache der vCPU-Zeit als ein einfacher SET/GET-Befehl benötigt und 3,2 KB Daten überträgt, verbraucht sie 3,2 ECPUs. Wenn sie dagegen nur 2 KB Daten überträgt, verbraucht sie 3 ECPUs.

ElastiCache Serverless gibt eine neue Metrik namens `awsElastiCacheProcessingUnits`, die Ihnen hilft, die von Ihrem Workload verbrauchten ECPUs zu verstehen.

Knotenstunden

Sie können einen eigenen Redis-Cache-Cluster entwerfen, indem Sie die EC2-Knotenfamilie, die Größe, die 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 für Redis-Backups

Ein Backup ist eine point-in-time Kopie eines Redis-Cache. ElastiCache ermöglicht es Ihnen, jederzeit ein Backup 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 Redis-Befehle](#).

Serverless-Caching

Amazon ElastiCache Serverless vereinfacht die Cache-Erstellung und skaliert sofort, um die anspruchsvollsten Anwendungen der Kunden zu unterstützen. Mit ElastiCache Serverless können Sie einen hochverfügbaren und skalierbaren Cache in weniger als einer Minute erstellen, sodass keine Cache-Cluster-Kapazität bereitgestellt, geplant und verwaltet werden muss. ElastiCache Serverless speichert Daten automatisch redundant über drei Availability Zones und bietet ein Service Level Agreement (SLA) für eine Verfügbarkeit von 99,99 %. Backups sind querkompatibel und können in selbst entwickelte Cluster exportiert und aus diesen wiederhergestellt werden.

Selbst entworfene Cluster

Wenn Sie eine differenzierte Kontrolle über Ihren ElastiCache für Redis-Cluster benötigen, können Sie Ihren eigenen Redis-Cluster mit entwerfen ElastiCache. ElastiCache ermöglicht Ihnen den Betrieb eines knotenbasierten Clusters, indem Sie den Knotentyp, die Anzahl der Knoten und die Knotenplatzierung über AWS Availability Zones für Ihren Cluster hinweg auswählen. Da ein vollständig verwalteter Service ElastiCache ist, hilft er bei der Verwaltung von Hardwarebereitstellung, Überwachung, Knotenaustausch und Software-Patching für Ihren Cluster. Selbst entwickelte Cluster können so konzipiert werden, dass sie ein SLA für die Verfügbarkeit von bis zu 99,99 % bieten. Backups sind querkompatibel und können in serverless-Caches exportiert und wiederhergestellt werden.

Auswählen aus den Bereitstellungsoptionen

Wählen Sie Serverless-Caching in folgenden Fällen:

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

Wählen Sie aus, ob Sie Ihren eigenen ElastiCache Cluster entwerfen möchten, wenn:

- Sie führen ElastiCache Serverless bereits aus und möchten eine genauere Kontrolle über den Knotentyp, auf dem Redis ausgeführt wird, die Anzahl der Knoten und die Platzierung von Knoten.

- Sie erwarten, dass Ihr Anwendungsdatenverkehr relativ vorhersehbar ist, und Sie möchten eine differenzierte Kontrolle über Leistung, Verfügbarkeit und Kosten.
- Sie können Ihre Kapazitätsanforderungen vorhersagen, um Kosten zu kontrollieren.

Vergleich von Serverless-Caching und selbst entwickelten Clustern

| Funktion | Serverless-Caching | Selbst entworfene Cluster |
|---|---|---|
| Cache-Einrichtung | Erstellen eines Cache mit nur einem Namen in weniger als einer Minute | Bietet eine differenzierte Kontrolle über das Cache-Cluster-Design. Benutzer können den Knotentyp, die Anzahl der Knoten und die Platzierung über AWS Availability Zones hinweg auswählen |
| Unterstützt ElastiCache für Redis-Version | ElastiCache für Redis Version 7.1 und höher | ElastiCache für Redis Version 4.0 und höher |
| Clustermodus | Betreiben von Redis <code>cluster mode enabled</code> nur in . Redis-Clients müssen unterstützen <code>cluster mode enabled</code> , um eine Verbindung zu ElastiCache Serverless herzustellen. | Kann für den Betrieb im aktivierten Clustermodus oder im deaktivierten Clustermodus konfiguriert werden. |
| Skalierung | Skaliert sowohl vertikal als auch horizontal ohne Kapazitätsmanagement. | Bietet die Kontrolle über die Skalierung und erfordert gleichzeitig eine Überwachung, um sicherzustellen, dass die aktuelle Kapazität den Bedarf angemessen erfüllt. Sie können die vertikale Skalierung wählen, indem Sie die Cache-Knotengröße |

| Funktion | Serverless-Caching | Selbst entworfene Cluster |
|--------------------|--|---|
| | | <p>bei Bedarf erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie Ihren Shards neue Shards hinzufügen oder weitere Replikate hinzufügen.</p> <p>Mit der Auto-Scaling-Funktion können Sie die Skalierung auch nach einem Zeitplan konfigurieren oder auf der Grundlage von Metriken wie CPU- und Arbeitsspeichernutzung im Cache skalieren.</p> |
| Client-Verbindung | Clients stellen eine Verbindung zu einem einzelnen Endpunkt her. Auf diese Weise kann sich die zugrundeliegende Cache-Knoten-Topologie (Skalierung, Ersatz und Upgrades) ändern, ohne den Client zu trennen. | Clients stellen eine Verbindung zu jedem einzelnen Cache-Knoten her. Wenn ein Knoten ersetzt wird, erkennt der Client die Cluster-Topologie neu und stellt Verbindungen wieder her. |
| Konfigurierbarkeit | Es ist keine detaillierte Konfiguration verfügbar. Kunden können grundlegende Einstellungen konfigurieren, einschließlich Subnetze, die auf den Cache zugreifen können, ob automatische Backups aktiviert oder deaktiviert sind und maximale Cache-Nutzungslimits. | Selbstentwerfene Cluster bieten detaillierte Konfigurationsoptionen. Kunden können Parametergruppen für eine differenzierte Steuerung verwenden. Eine Tabelle dieser Parameterwerte nach Knotentyp finden Sie unter Spezifische Parameter für den Redis-Knotentyp . |

| Funktion | Serverless-Caching | Selbst entworfene Cluster |
|---|---|---|
| Multi-AZ | Daten werden asynchron über mehrere Availability Zones repliziert, um eine höhere Verfügbarkeit und eine verbesserte Leselatenz zu gewährleisten. | Bietet die Möglichkeit, den Cluster in einer einzelnen Availability Zone oder über mehrere Availability Zones (AZs) hinweg zu entwerfen. Bei Multi-AZ-Clustern werden Daten asynchron über mehrere Availability Zones repliziert, um die Verfügbarkeit zu erhöhen und die Leselatenz zu verbessern. |
| 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 während der Übertragung (TLS) | Immer aktiviert. Clients müssen TLS-Konnektivität unterstützen. | Option zum Aktivieren oder Deaktivieren. |

| Funktion | Serverless-Caching | Selbst entworfene Cluster |
|-------------|---|--|
| Sicherungen | <p>Unterstützt automatische und manuelle Backups von Caches ohne Leistungs einbußen.</p> <p>Backups sind übergreifend und können in einem - ElastiCache Serverless-Cache oder einem selbst entwickelten Cluster wiederhergestellt werden.</p> | <p>Unterstützt automatische und manuelle Backups. Cluster können je nach verfügbar em reserviertem Speicher einige Leistungseinbußen sehen. Weitere Informationen finden Sie unter Verwalten von reserviertem Speicher.</p> <p>Backups sind übergreifend und können in einem - ElastiCache Serverless-Cache oder einem selbst entwickelten Cluster wiederhergestellt werden.</p> |
| Überwachen | <p>Unterstützt Metriken auf Cache-Ebene, einschließlich Cache-Trefferrate, Cache-Fehlerrate, Datengröße und verbrauchter ECPUs.</p> <p>ElastiCache Serverless sendet Ereignisse mit EventBridge , wenn wichtige Ereignisse in Ihrem Cache auftreten. Mit Amazon können Sie ElastiCache Ereignisse überwachen, aufnehmen, transformieren und entsprechend handeln EventBridge. Weitere Informationen finden Sie unter Serverless-Cache-Ereignisse.</p> | <p>ElastiCache Selbst entwickelte Cluster geben auf jeder Knotenebene Metriken aus, einschließlich Metriken auf Host- und Cache-Ebene.</p> <p>Selbst entwickelte Cluster geben SNS-Benachrichtigungen für wichtige Ereignisse aus. Siehe Metriken für Redis.</p> |

| Funktion | Serverless-Caching | Selbst entworfene Cluster |
|-------------------------------|---|---|
| Verfügbarkeit | Service Level Agreement (SLA) mit 99,99 % Verfügbarkeit | Selbst entwickelte Cluster können so konzipiert werden, dass je nach Konfiguration ein Service Level Agreement (SLA) von bis zu 99,99 % verfügbar ist. |
| Software-Upgrades und Patches | Aktualisiert die Cache-Software automatisch auf die neueste Neben- und Patch-Version, ohne dass die Anwendung Auswirkungen hat. Kunden erhalten eine Benachrichtigung über Hauptversions-Upgrades, und Kunden können bei Bedarf auf die neueste Hauptversion aktualisieren. | Selbstentwickelte Cluster bieten einen kundenfähigen Self-Service für Neben- und Patching-Versionsupgrades sowie Hauptversionsupgrades. Verwaltete Updates werden während kundendefinierter Wartungsfenster automatisch angewendet. Kunden können auch ein Neben- oder Patch-Versions-Upgrade auf Abruf anwenden. |
| Globaler Datenspeicher | Nicht unterstützt | Unterstützt den Global Data Store, der die regionsübergreifende Replikation mit Schreibvorgängen in einer einzelnen Region und Lesevorgängen in mehreren Regionen ermöglicht |

| Funktion | Serverless-Caching | Selbst entworfene Cluster |
|---------------|--|--|
| Daten-Tiering | Nicht unterstützt | Cluster, die mit Knoten aus der R6gd-Familie entwickelt wurden, haben ihre Daten zwischen Arbeitsspeicher und lokalem SSD-Speicher (Solid State Drives). Daten-Tiering bietet eine Preis-Leistungs-Option für Redis-Workloads, indem neben dem Speichern von Daten im Speicher 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 Anforderungen in ElastiCache Processing Units (ECPU). Weitere Informationen zu Preisen finden Sie hier . | Pay-per-hour, basierend auf der Cache-Knotennutzung. Weitere Informationen zu Preisen finden Sie hier . |

Verwandte Themen:

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

Amazon-ElastiCache-Ressourcen

Wir empfehlen Ihnen, mit der Lektüre der folgenden Abschnitte zu beginnen und bei Bedarf darauf zurückzugreifen:

- Service-Merkmale und Preise – Die [Produktdetailseite](#) bietet eine allgemeine Produktübersicht über ElastiCache, Service-Merkmale und Preise.

- [ElastiCache-Videos](#) – Im [ElastiCache-Videos](#)-Abschnitt finden Sie Videos, die Sie in Amazon ElastiCache einführen. Die Videos behandeln häufige Anwendungsfälle für ElastiCache und demonstrieren, wie ElastiCache zur Verringerung der Latenz und zur Verbesserung des Durchsatzes für Ihre Anwendungen eingesetzt werden kann.
- [Erste Schritte](#) – Der [Erste Schritte mit Amazon ElastiCache for Redis](#)-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 Whitepaper [Skalierbare Leistung mit Amazon ElastiCache](#) befasst sich mit Caching-Strategien, die Ihrer Anwendung zu einer guten skalierbaren Leistung verhelfen.

Wenn Sie das AWS Command Line Interface (AWS CLI) verwenden möchten, beginnen Sie am besten mit diesen Dokumenten:

- [AWS Command Line Interface-Dokumentation](#)

In diesem Abschnitt finden Sie Informationen zum Herunterladen vom AWS CLI, zur Inbetriebnahme des AWS CLI auf Ihrem System und zur Bereitstellung Ihrer AWS-Anmeldedaten.

- [AWS CLI-Dokumentation für ElastiCache](#)

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

Sie können Anwendungsprogramme schreiben, um die ElastiCache-API mit einer Vielzahl von gängigen Programmiersprachen zu nutzen. Nachfolgend finden Sie einige Ressourcen:

- [Tools für Amazon Web Services](#)

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

- [Verwenden der ElastiCache-API](#)

Falls Sie die AWS-SDKs nicht verwendet möchten, können Sie mit ElastiCache direkt über die Abfrage-API kommunizieren. In diesem Abschnitt finden Sie auch Tipps zur Fehlerbehebung

sowie Informationen zur Erstellung und Authentifizierung von Anfragen und der Behandlung von Antworten.

- [Amazon-ElastiCache-API-Referenz](#)

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

AWS-Regionen und Availability Zones

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 eindeutige Standorte, die Verfügbarkeitszonen oder AZs genannt werden. Jede Availability Zone ist so aufgebaut, dass sie von Fehlern in anderen Availability Zones nicht betroffen ist. Jede ist so aufgebaut, dass sie anderen Availability Zones in der gleichen AWS-Region kostengünstige Netzwerkkonnektivität mit geringer Latenz zur Verfügung stellt. 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 [Minimieren von Ausfallzeiten in ElastiCache for Redis mit Multi-AZ](#).

Häufige Anwendungsfälle von ElastiCache und wie ElastiCache helfen kann

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 eine optimierte Datenbankabfrage oder ein Remote-API-Aufruf ist deutlich langsamer als das Abrufen eines flachen Schlüssels aus einem In-Memory-Cache. Eine merkliche Verlangsamung führt dazu, dass Kunden abwandern.

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

Themen

- [In-Memory-Datastore](#)
- [Gaming Leaderboards \(Sortierte Sätze von Redis\)](#)
- [Messaging \(Redis Pub/Sub\)](#)
- [Empfehlungsdaten \(Redis-Hashes\)](#)
- [Andere Verwendung von Redis](#)
- [Kundenstimmen zu ElastiCache](#)

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 interessantesten 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 hier:

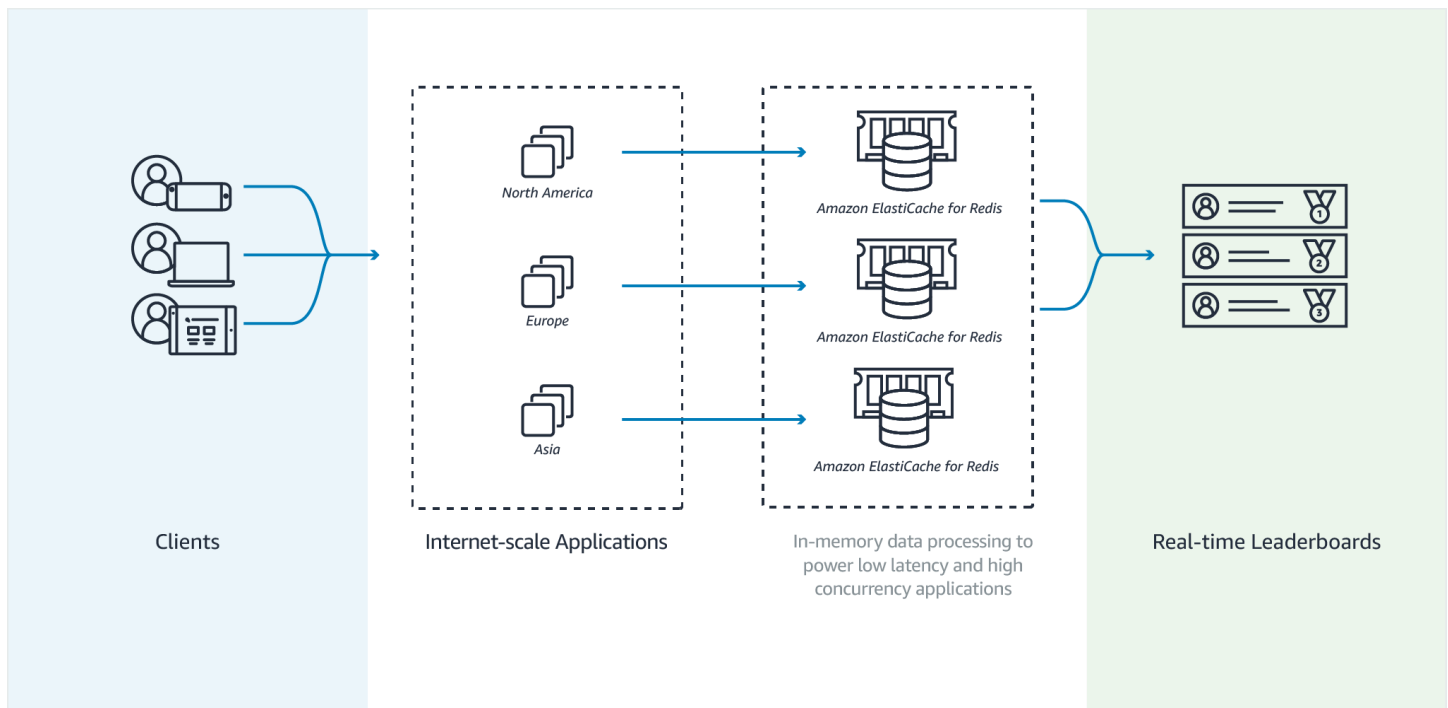
- [Caching-Strategien](#) im Benutzerhandbuch zu ElastiCache for Redis

Gaming Leaderboards (Sortierte Sätze von Redis)

Redis-sortierte-Sets verlagern die Rechenkomplexität von Ranglisten von Ihrer Anwendung zu Ihrem Redis-Cluster.

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. Sorted Sets von Redis garantieren sowohl Eindeutigkeit als auch die Sortierung der Elemente. Mit sortierten Sätzen von Redis wird bei jedem Hinzufügen eines neuen Elements zum sortierten Satz dieser in Echtzeit neu geordnet. Es wird dann in der numerisch richtigen Reihenfolge in die Menge eingefügt.

Im folgenden Diagramm können Sie sehen, wie eine ElastiCache-for-Redis-Gaming-Rangliste funktioniert.



Example – Redis Leaderboard

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 die Rangliste auf der Basis von Nullen erstellt wird, ergibt ZREVRANK eine 1 für June, die an zweiter Stelle steht.

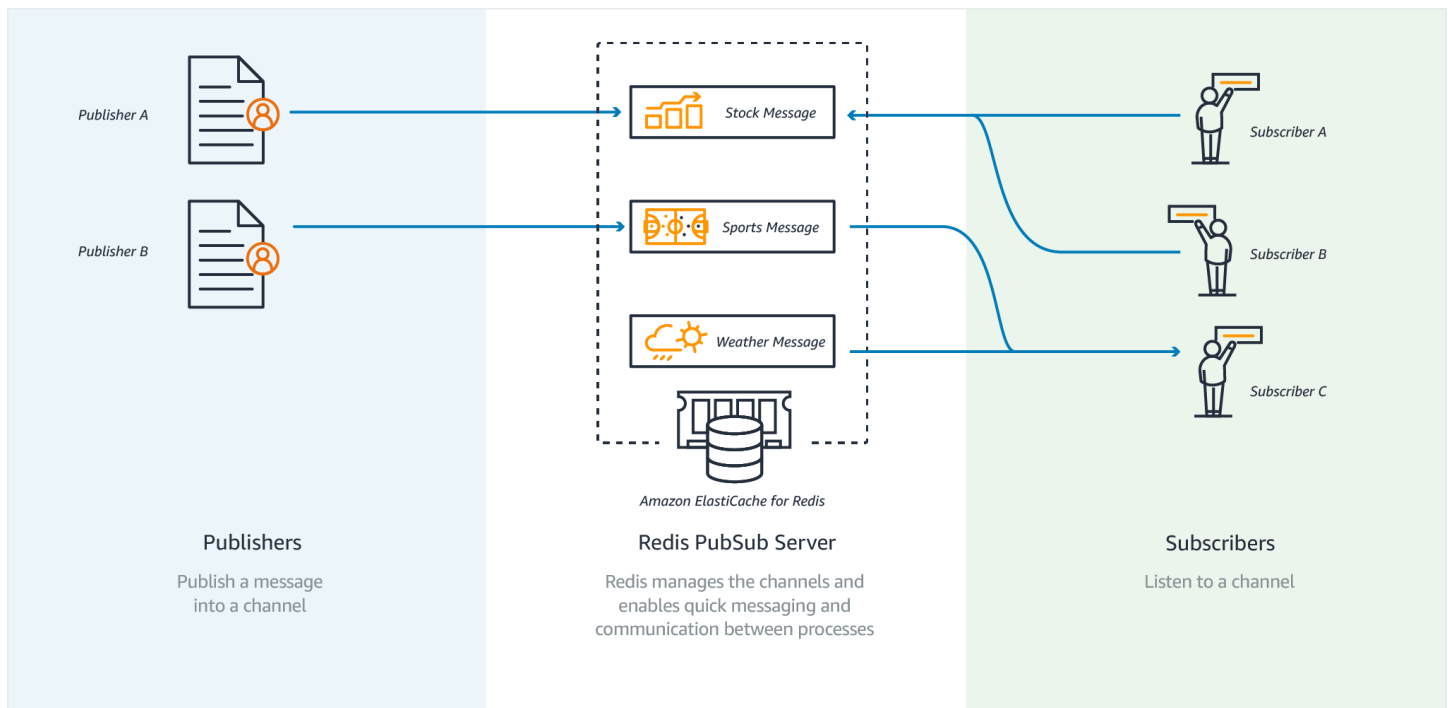
```
ZREVRANK leaderboard June
1
```

Weitere Informationen finden Sie in der [Redis-Dokumentation](#) über sortierte Sets.

Messaging (Redis Pub/Sub)

Wenn Sie eine E-Mail-Nachricht versenden, senden Sie sie an mindestens einen angegebenen Empfänger. Im Pub-/Sub-Paradigma senden Sie eine Nachricht an einen bestimmten Channel, ohne zu wissen wer, wenn überhaupt, die Nachricht erhalten wird. 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.

Pub/Sub-Funktionalität von Redis hat keinen Bezug zu einem Schlüsselraum. Daher gibt es auf keinem Level Beeinträchtigungen. In der folgenden Abbildung finden Sie eine Illustration von ElastiCache-for-Redis-Nachrichten.



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 Channel zu abonnieren, verwenden Sie den `SUBSCRIBE`-Befehl unter Angabe des Channels, 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 das Abonnement für den Channel mithilfe des `UNSUBSCRIBE`-Befehls unter Angabe des Channels, dessen Abonnement gekündigt werden soll.

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


Example – Abonnement mehrerer ausgewählter Channels

Um mehrere bestimmte Channels zu abonnieren, listen Sie die Channels 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 Befehl UNSUBSCRIBE 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 Befehl UNSUBSCRIBE und geben Sie die Kanäle an, die Sie abbestellen 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

Clients können alle Channels abonnieren, die einem Muster entsprechen, indem sie den PSUBSCRIBE-Befehl verwenden.

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 Channel-Zeichenfolgen, die an einen [P]SUBSCRIBE-Befehl bzw. an den [P]UNSUBSCRIBE-Befehl gesendet werden, müssen 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 in einem Kanal veröffentlichen, den er abonniert hat.

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

Empfehlungsdaten (Redis-Hashes)

Die Verwendung von INCR oder DECR in Redis macht Kompilierungsempfehlungen einfach. Immer dann, wenn Benutzer ein Produkt "likern", erhöhen Sie einen item:productID:like-Zähler. Immer dann, wenn Benutzer ein Produkt ein "dislikern", erhöhen Sie einen item:productID:dislike-Zähler. Mithilfe von Redis-Hashes können Sie außerdem eine Liste aller Personen führen, denen ein Produkt gefällt (Like) oder nicht gefällt (Dislike).

Example – Likes und Dislikes

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

HSET item:38923:ratings Tommy -1

Andere Verwendung von Redis

Der Blog-Beitrag [Wie man Redis nutzen kann, indem man es einfach zu seinem Stack hinzufügt](#) von Salvatore Sanfilippo bespricht eine Reihe von häufigen Datenbankproblemen und wie sie mit Redis einfach gelöst werden können. Dieser Ansatz entlastet Ihre Datenbank und verbessert die Leistung.

Kundenstimmen zu ElastiCache

Wenn Sie erfahren möchten, wie Unternehmen wie Airbnb, PBS, Esri und andere Amazon ElastiCache nutzen, um ihr Geschäft mit einer verbesserten Kundenerfahrung auszubauen, lesen Sie bitte [Wie andere Amazon ElastiCache nutzen](#).

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

Erste Schritte mit Amazon ElastiCache for Redis

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

Themen

- [Einrichtung](#)
- [Schritt 1: Erstellen eines Caches](#)
- [Schritt 2: Daten lesen und in den Cache schreiben](#)
- [Schritt 3: \(Optional\) Bereinigen](#)
- [Nächste Schritte](#)
- [Erste Schritte mit ElastiCache und AWS-SDKs](#)
- [Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC](#)

Einrichtung

Zum Einrichten ElastiCache:

Themen

- [Melde dich an für ein AWS-Konto](#)
- [Erstellen Sie einen Benutzer mit Administratorzugriff](#)
- [Erteilen programmgesteuerten Zugriffs](#)
- [Richten Sie Ihre Berechtigungen ein \(nur ElastiCache für neue Benutzer\)](#)
- [Einrichten von EC2](#)
- [Gewähren des Netzwerkzugriffs aus einer Amazon-VPC-Sicherheitsgruppe für den Cache](#)
- [Herunterladen und Einrichten von redis-cli](#)

Melde dich 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/signup>.
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. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um [Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich](#) ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Erstellen Sie einen Benutzer 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 den Root-Benutzer.

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

Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das 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.

Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-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](#).

Weisen Sie weiteren Benutzern Zugriff zu

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

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

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

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

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 von der Art des Benutzers 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>(Benutzer, die in IAM Identity Center verwaltet werden)</p> | <p>Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS</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 zu AWS IAM Identity Center verwendenden im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs, Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch für AWS SDKs und Tools. |
| IAM | <p>Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS</p> | <p>Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.</p> |
| IAM | <p>(Nicht empfohlen)</p> <p>Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an die AWS CLI, AWS SDKs oder APIs zu signieren. AWS</p> | <p>Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.</p> <ul style="list-style-type: none"> • Informationen dazu finden Sie unter Authentifizierung mithilfe von IAM-Benutzeranmeldedaten im Benutzerhandbuch. AWS |

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

Verwandte Themen:

- [Was ist IAM](#) im IAM-Benutzerhandbuch.
- AWS Allgemeine Referenz zu [Sicherheitsanmeldedaten](#).

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, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter [Erstellen einer Rolle für einen externen Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

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

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:

- [Erstellen einer neuen Richtlinie](#)(IAM)
- [AWS-verwaltete Richtlinien für Amazon ElastiCache](#)
- [Verwenden von serviceverknüpften Rollen für Amazon ElastiCache](#)

Einrichten von EC2

Sie müssen eine EC2-Instance einrichten, über die Sie eine Verbindung mit Ihrem Cache herstellen.

- Wenn Sie noch keine EC2-Instance haben, erfahren Sie hier, wie Sie eine EC2-Instance einrichten: [Erste Schritte mit EC2](#).
- Ihre EC2-Instance muss sich in derselben VPC wie Ihr Cache befinden und dieselben Sicherheitsgruppeneinstellungen haben. Standardmäßig ElastiCache erstellt Amazon einen Cache in Ihrer Standard-VPC und verwendet die Standardsicherheitsgruppe. Damit Sie diesem Tutorial folgen können, stellen Sie sicher, dass sich Ihre EC2-Instance in der Standard-VPC befindet und über die Standardsicherheitsgruppe verfügt.

Gewähren des Netzwerkzugriffs aus einer Amazon-VPC-Sicherheitsgruppe für den Cache

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

1. Melden Sie sich bei der [Amazon EC2 EC2-Konsole](#) an AWS Command Line Interface und öffnen Sie sie.
2. Wählen Sie in der Navigationsleiste unter Network & Security die Option Security Groups aus.
3. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe Ihrer Amazon VPC aus. 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. Geben Sie in das Feld Portbereich den Wert 6379 ein.
 - e. Wählen Sie im Feld Quelle die Option Überall mit dem Portbereich (0.0.0.0/0), damit jede Amazon-EC2-Instance, die Sie innerhalb Ihrer Amazon VPC starten, eine Verbindung mit Ihrem 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. Geben Sie in das Feld Portbereich den Wert 6380 ein.
 - i. Wählen Sie im Feld Quelle die Option Überall mit dem Portbereich (0.0.0.0/0), damit jede Amazon-EC2-Instance, die Sie innerhalb Ihrer Amazon VPC starten, eine Verbindung mit Ihrem Cache herstellen kann.
 - j. Wählen Sie Speichern.

Herunterladen und Einrichten von redis-cli

1. Verbinden Sie sich mit Ihrer Amazon-EC2-Instance mit dem Verbindungsdienstprogramm Ihrer Wahl. Anweisungen zur Verbindung mit einer Amazon-EC2-Instance finden Sie im [Amazon-EC2-Erste-Schritte-Handbuch](#).
2. Laden Sie das redis-cli-Dienstprogramm herunter und installieren Sie es mit dem entsprechenden Befehl für Ihre Einrichtung.

Amazon Linux 2023

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

Amazon Linux 2

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

Note

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

Schritt 1: Erstellen eines Caches

In diesem Schritt wird ein neuer Cache in Amazon ElastiCache erstellt.

AWS Management Console

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

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Konsole unter <https://console.aws.amazon.com/connect/>.
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Redis-Caches.
3. Wählen Sie auf der rechten Seite der Konsole Redis-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 sich der Cache im Status „AKTIV“ befindet, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

AWS CLI

Im folgenden AWS CLI-Beispiel wird mithilfe von `create-serverless-cache` ein neuer Cache erstellt.

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.

Überprüfen Sie mit dem Befehl `describe-serverless-caches`, ob ElastiCache den Cache fertig erstellt hat.

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 [Schritt 2: Daten lesen und in den Cache schreiben](#) fort.

Schritt 2: Daten lesen und in den Cache schreiben

In diesem Abschnitt wird davon ausgegangen, dass Sie eine Amazon-EC2-Instance erstellt haben und eine Verbindung mit ihr möglich ist. Anweisungen dazu finden Sie im [Amazon-EC2-Erste-Schritte-Leitfaden](#).

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

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 Redis-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
```

Herstellen einer Verbindung mit Ihrem Redis-Cache (Linux)

Nachdem Sie den gewünschten Endpunkt gefunden haben, können Sie sich bei einer EC2-Instance anmelden und eine Verbindung mit dem Cache herstellen. Im folgenden Beispiel verwenden Sie das `redis-cli`-Dienstprogramm, um eine Verbindung mit 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/redis-cli -h cache-endpoint --tls -p 6379  
set a "hello"           // Set key "a" with a string value and no expiration  
OK  
get a                   // Get value for key "a"  
"hello"
```

Herstellen einer Verbindung mit Ihrem Redis-Cache (Windows)

Nachdem Sie den gewünschten Endpunkt gefunden haben, können Sie sich bei einer EC2-Instance anmelden und eine Verbindung mit dem Cache herstellen. Im folgenden Beispiel verwenden Sie das `redis-cli`-Dienstprogramm, um eine Verbindung mit einem Cluster herzustellen. Mit dem folgenden Befehl stellen Sie eine Verbindung mit einem Cache her. Öffnen Sie die Befehlszeile, wechseln Sie in das Redis-Verzeichnis und führen Sie den Befehl aus. (Hinweis: Ersetzen Sie `Cache_Endpoint` durch den Endpunkt, den Sie im vorherigen Schritt abgerufen haben.)

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

Sie können jetzt mit [Schritt 3: \(Optional\) Bereinigen](#) fortfahren.

Schritt 3: (Optional) Bereinigen

Wenn Sie den erstellten Amazon-ElastiCache-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. Zum Löschen Ihres Caches können Sie die ElastiCache-Konsole, die AWS CLI oder die ElastiCache-API verwenden.

AWS Management Console

So löschen Sie Ihren Cache mithilfe der Konsole:

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Redis-Caches.
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 in den Status DELETING wechselt, fallen für ihn keine Gebühren mehr an.

AWS CLI

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

Linux

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

Windows

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

Beachten Sie, dass für das Feld Status der Wert DELETING festgelegt ist.

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

Nächste Schritte

Weitere Informationen zu ElastiCache finden Sie auf den folgenden Seiten:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache für Redis](#)
- [Protokollierung und Überwachung in Amazon ElastiCache](#)
- [ElastiCache Best Practices und Caching-Strategien](#)
- [Snapshot und Wiederherstellung](#)
- [Amazon-SNS-Überwachung von ElastiCache-Ereignissen](#)

Erste Schritte mit ElastiCache und AWS-SDKs

Dieser Abschnitt enthält praktische Tutorials, die Ihnen helfen, Amazon ElastiCache besser kennenzulernen. Wir empfehlen Ihnen, eines der sprachspezifischen Tutorials durchzuarbeiten.

Note

Die AWS SDKs stehen für eine Vielzahl von Sprachen zur Verfügung. Eine vollständige Liste finden Sie unter [Tools für Amazon Web Services](#).

Python und ElastiCache

In diesem Tutorial verwenden Sie AWS-SDK-for-Python (Boto 3) für das Schreiben einfacher Programme, um die folgenden ElastiCache-Operationen durchzuführen:

- Erstellen von ElastiCache-Clustern (Cluster-Modus aktiviert und Cluster-Modus deaktiviert)
- Überprüfen Sie, ob Benutzer oder Benutzergruppen vorhanden sind, ansonsten erstellen Sie sie (nur ab Redis 6.0)
- Verbinden zu 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 des AWS-SDK-for-Python (Boto) beziehen. Der folgende Abschnitt ist spezifisch für ElastiCache: [ElastiCache-Low-Level-Client](#)

Tutorial-Voraussetzungen

- Richten Sie einen AWS-Zugriffsschlüssel für die Verwendung der AWS-SDKs ein. Weitere Informationen finden Sie unter [Einrichtung](#).
- 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.

ElastiCache Cluster und Benutzer erstellen

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

Themen

- [Erstellen Sie einen Cluster mit deaktiviertem Cluster-Modus](#)
- [Erstellen eines Clusters mit deaktivierten Cluster-Modus mit TLS und RBAC](#)
- [Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus](#)
- [Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus mit TLS und 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 mit dem Namen `.py` ein.

CreateClusterModeDisabledCluster

```
import boto3
import logging

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

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

Returns a dictionary with the API response

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

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

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

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

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

:param ReplicationGroupDescription: Description for the cluster.

:param ReplicationGroupId: Name for the cluster

:return: dictionary with the API results

```
"""
```

```
if not ReplicationGroupId:
```

```
    return 'ReplicationGroupId parameter is required'
```

```
response = client.create_replication_group(
```

```
    AutomaticFailoverEnabled=True,
```

```
    CacheNodeType=CacheNodeType,
```

```
    Engine='redis',
```

```
    EngineVersion=EngineVersion,
```

```
    NumCacheClusters=NumCacheClusters,
```

```
    ReplicationGroupDescription=ReplicationGroupDescription,
```

```
    ReplicationGroupId=ReplicationGroupId,
```

```
    SnapshotRetentionLimit=30,
```

```
)
```

```
return response
```

```
if __name__ == '__main__':
```

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

```
    elasticacheResponse = create_cluster_mode_disabled(
```

```
        #CacheNodeType='cache.m6g.large',
```

```
        EngineVersion='6.0',
```

```
        NumCacheClusters=3,
```

```
        ReplicationGroupDescription='Redis cluster mode disabled with replicas',
```

```
        ReplicationGroupId='redis202104053'
```

```
    )
```

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

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

```
python CreateClusterModeDisabledCluster.py
```

Weitere Informationen finden Sie unter [Verwalten von Clustern](#).

Erstellen eines Clusters mit deaktiviertem Cluster-Modus mit TLS und RBAC

Um die Sicherheit zu gewährleisten, können Sie Transport Layer Security (TLS) und rollenbasierte Zugriffssteuerung (RBAC) beim Erstellen eines Clusters mit deaktiviertem Cluster-Modus verwenden. Im Unterschied zu Redis-AUTH, bei dem alle authentifizierten Clients vollen Zugriff auf Replikationsgruppen haben, wenn ihr Token authentifiziert ist, können Sie mit RBAC 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 `ClusterModeDisabledWithRBAC.py` ein.

```
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', NumCacheNodes=1,
    cache_cluster', ReplicationGroupId=None, UserGroupIds=None,
    SecurityGroupIds=None, CacheSubnetGroupName=None):
    """Creates an ElastiCache Cluster with cluster mode disabled and RBAC

    Returns a dictionary with the API response

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

```

:param NumCacheClusters: Number of nodes in the cluster. Minimum 1 (just a primary
node) and maximum 6 (1 primary and 5 replicas).
If not specified, cluster will be created with 1 primary and 1 replica.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Mandatory name for the cluster.
:param UserGroupIds: The ID of the user group to be assigned to the cluster.
:param SecurityGroupIds: List of security groups to be assigned. If not defined,
default will be used
:param CacheSubnetGroupName: subnet group where the cluster will be placed. If not
defined, default will be used.
:return: dictionary with the API results

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

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

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

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

if __name__ == '__main__':

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

```

```
response=create_cluster_mode_disabled_rbac(
    CacheNodeType='cache.m6g.large',
    EngineVersion='6.0',
    NumCacheClusters=3,
    ReplicationGroupDescription='Redis cluster mode disabled with replicas',
    ReplicationGroupId='redis202104',
    UserGroupIds=[
        'mygroup'
    ],
    SecurityGroupIds=[
        'sg-7cc73803'
    ],
    CacheSubnetGroupName='default'
)

logging.info(response)
```

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

```
python ClusterModeDisabledWithRBAC.py
```

Weitere Informationen finden Sie unter [Verwalten von Clustern](#).

Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus

Kopieren Sie das folgende Programm und fügen Sie es in eine Datei mit dem Namen `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/red-ug/CacheNodes.SupportedTypes.html for supported node types
:param EngineVersion: Engine version to be used. If not specified, latest will be
used.
:param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
If not specified, cluster will be created with 1 shard.
:param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
replica per shard will be created.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Name for the cluster
:return: dictionary with the API results

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

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

return response

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

```

```
    ReplicasPerNodeGroup=1,  
)  
  
logging.info(response)
```

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

```
python ClusterModeEnabled.py
```

Weitere Informationen finden Sie unter [Verwalten von Clustern](#).

Erstellen Sie einen Cluster mit aktiviertem Cluster-Modus mit TLS und RBAC

Um die Sicherheit zu gewährleisten, können Sie Transport Layer Security (TLS) und rollenbasierte Zugriffssteuerung (RBAC) beim Erstellen eines Clusters mit aktiviertem Cluster-Modus verwenden. Im Unterschied zu Redis-AUTH, bei dem alle authentifizierten Clients vollen Zugriff auf Replikationsgruppen haben, wenn ihr Token authentifiziert ist, können Sie mit RBAC 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 `ClusterModeEnabledWithRBAC.py` ein.

```
import boto3  
import logging  
  
logging.basicConfig(level=logging.INFO)  
client = boto3.client('elasticsearch')  
  
def  
    create_cluster_mode_enabled(CacheNodeType='cache.t3.small', EngineVersion='6.0', NumNodeGroups=1,  
                                ReplicationGroupDescription='Sample cache with cluster  
mode enabled', ReplicationGroupId=None, UserGroupIds=None,  
                                SecurityGroupIds=None, CacheSubnetGroupName=None, CacheParameterGroupName='default.redis6.0.clus  
        """Creates an ElastiCache Cluster with cluster mode enabled and RBAC  
  
        Returns a dictionary with the API response  
  
        :param CacheNodeType: Node type used on the cluster. If not specified,  
cache.t3.small will be used  
        Refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/  
CacheNodes.SupportedTypes.html for supported node types
```

```
:param EngineVersion: Engine version to be used. If not specified, latest will be
used.
:param NumNodeGroups: Number of shards in the cluster. Minimum 1 and maximum 90.
If not specified, cluster will be created with 1 shard.
:param ReplicasPerNodeGroup: Number of replicas per shard. If not specified 1
replica per shard will be created.
:param ReplicationGroupDescription: Description for the cluster.
:param ReplicationGroupId: Name for the cluster.
:param CacheParameterGroupName: Parameter group to be used. Must be compatible with
the engine version and cluster mode enabled.
:return: dictionary with the API results

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

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

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

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

if __name__ == '__main__':
    # Creates a cluster mode enabled cluster
    response = create_cluster_mode_enabled(
```



```
CacheNodeType='cache.m6g.large',
EngineVersion='6.0',
ReplicationGroupDescription='Redis cluster mode enabled with replicas',
ReplicationGroupId='redis2021',
# Creates a cluster mode enabled cluster with 1 shard(NumNodeGroups), 1 primary
(implicit) and 2 replicas (replicasPerNodeGroup)
NumNodeGroups=2,
ReplicasPerNodeGroup=1,
UserGroupIds=[
    'mygroup'
],
SecurityGroupIds=[
    'sg-7cc73803'
],
CacheSubnetGroupName='default'

)

logging.info(response)
```

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

```
python ClusterModeEnabledWithRBAC.py
```

Weitere Informationen finden Sie unter [Verwalten von Clustern](#).

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

Mithilfe von RBAC erstellen Sie Benutzer und weisen ihnen spezifische Berechtigungen zu, indem Sie eine Zugriffszeichenfolge verwenden, wie im Folgenden beschrieben. Sie weisen die Benutzer Benutzergruppen zu, denen eine bestimmte Rolle zugewiesen ist (Administratoren, Personalabteilung), die dann einer oder mehreren ElastiCache Redis-Replikationsgruppen zugewiesen werden. Dadurch können Sie Sicherheitsgrenzen zwischen Clients einrichten, die dieselbe(n) Redis-Replikationsgruppe(n) verwenden, und verhindern, dass Clients gegenseitig auf ihre 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.

```
import boto3
import logging

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

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

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

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

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

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

```
        return False
    else:
        raise

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

    Returns the ARN for the newly created user or the error message
    :param UserId: ElastiCache user ID. User IDs must be unique
    :param Username: ElastiCache user name. ElastiCache allows multiple users with the
    same name as long as the associated user ID is unique.
    :param Password: Password for user. Must have at least 16 chars.
    :param AccessString: Access string with the permissions for the user. For
    details refer to https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.RBAC.html#Access-string
    :return: user ARN
    """
    try:
        response = client.create_user(
            UserId=UserId,
            Username=Username,
            Engine='Redis',
            Passwords=[Password],
            AccessString=AccessString,
            NoPasswordRequired=False
        )
        return response['ARN']
    except Exception as e:
        logging.info(e.response['Error'])
        return e.response['Error']

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

    Return: Group ARN
    :param UserIds: List with user IDs to be associated with the new group. A default
    user is required
    :param UserGroupId: The ID (name) for the group
    :return: Group ARN
    """
    try:
        response = client.create_user_group(
            UserGroupId=UserGroupId,
```

```
        Engine='Redis',
        UserIds=UserIds
    )
    return response['ARN']
except Exception as e:
    logging.info(e.response['Error'])

if __name__ == '__main__':

    groupName='mygroup2'
    userName = 'myuser2'
    userId=groupName+'-'+userName

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

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

```
python UserAndUserGroups.py
```

Herstellen einer Verbindung mit ElastiCache

Die folgenden Beispiele verwenden den Redis-Client, um eine Verbindung mit ElastiCache herzustellen.

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 mit dem Namen `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

Die folgenden Beispiele verwenden das boto3-SDK-für-ElastiCache, um mit ElastiCache zu arbeiten.

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 mit dem Namen `SetAndGetStrings.py` ein.

```
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 mit dem Namen `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 mit dem Namen `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.psubsub()
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 mit dem Namen `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 acknowleges the message, it won't be provided again. If message
    wasn't acknowledged, it can be served to another thread.

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

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

```

```
        except:
            raise

        time.sleep(0.5)

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

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

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

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

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

```
python ReadWriteStream.py
```

Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon 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](#)
- [Schritt 2: Erstellen einer Lambda-Funktion](#)
- [Schritt 3: Testen der Lambda -Funktion](#)
- [Schritt 4: Aufräumen \(optional\)](#)

Schritt 1: Einen serverlosen Cache erstellen

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

Themen

- [Schritt 1.1: Erstellen Sie einen serverlosen Cache](#)
- [Schritt 1.2: Kopieren Sie den serverlosen Cache-Endpunkt](#)
- [Schritt 1.3: IAM-Rolle erstellen](#)
- [Schritt 1.4: Erstellen Sie einen serverlosen Cache](#)

Schritt 1.1: Erstellen Sie einen serverlosen Cache

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

[Schritt 1: Erstellen eines Caches](#)

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

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-Endpunkt

Stellen Sie mit dem Befehl sicher, dass ElastiCache for Redis 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: IAM-Rolle erstellen

1. Erstellen Sie, wie unten dargestellt, ein Dokument mit den IAM-Vertrauensrichtlinien für Ihre Rolle, sodass Ihr Konto die neue Rolle übernehmen kann. 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": {
      "Service": "lambda.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

2. Erstellen Sie ein IAM-Richtliniendokument wie im Folgenden dargestellt. 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"
      ]
    }
  ]
}
```

```
}
```

3. Erstellen Sie eine IAM-Rolle.

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

4. Erstellen Sie die IAM-Richtlinie.

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

5. Fügen 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 einer Lambda-Funktion

Gehen Sie wie folgt vor, um eine Lambda-Funktion zu erstellen.

Themen

- [Schritt 2.1: Erstellen Sie eine Lambda-Funktion](#)
- [Schritt 2.2: Eine IAM-Rolle erstellen \(Ausführungsrolle\)](#)
- [Schritt 2.3: Aktualisieren des Bereitstellungspakets \(Erstellen der Lambda-Funktion\)](#)

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 in Schritt 1.2 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 = botocore.session.get_session()
        self.request_signer = RequestSigner(
            ServiceId("elasticache"),
            self.region,
            "elasticache",
            "v4",
            session.get_credentials(),
            session.get_component("event_emitter"),
        )

        # Generated IAM tokens are valid for 15 minutes
        @cached(cache=TTLCache(maxsize=128, ttl=900))
        def get_credentials(self) -> Union[Tuple[str], Tuple[str, str]]:
            query_params = {"Action": "connect", "User": self.user}
            if self.is_serverless:
                query_params["ResourceType"] = "ServerlessCache"
            url = urlunparse(
                ParseResult(
                    scheme="https",
                    netloc=self.cache_name,
                    path="/",
                    query=urlencode(query_params),
                    params="",
                    fragment="",
                )
            )
            signed_url = self.request_signer.generate_presigned_url(
                {"method": "GET", "url": url, "body": {}, "headers": {}, "context": {}},
                operation_name="connect",
                expires_in=900,
                region_name=self.region,
            )
            # RequestSigner only seems to work if the URL has a protocol, but
            # Elasticache only accepts the URL without a protocol
            # So strip it off the signed URL before returning
            return (self.user, signed_url.removeprefix("https://"))
```

```
def lambda_handler(event, context):
    username = "iam-user-01" # replace with your user id
    cache_name = "cache-01" # replace with your cache name
    elasticache_endpoint = "cache-01-xxxxx.serverless.us-east-1.cache.amazonaws.com" #
replace with your cache endpoint
    creds_provider = ElastiCacheIAMProvider(user=username, cache_name=cache_name,
is_serverless=True)
    redis_client = redis.Redis(host=elasticache_endpoint, port=6379,
credential_provider=creds_provider, ssl=True, ssl_cert_reqs="none")

    key='uuid'
    # create a random UUID - this will be the sample element we add to the cache
    uuid_in = uuid.uuid4().hex
    redis_client.set(key, uuid_in)
    result = redis_client.get(key)
    decoded_result = result.decode("utf-8")
    # check the retrieved item matches the item added to the cache and print
    # the results
    if decoded_result == uuid_in:
        print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Redis.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
{decoded_result}")

    return "Fetched value from Redis"
```

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 Quelldatei `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: Eine IAM-Rolle erstellen (Ausführungsrolle)

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

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

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

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

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

Verwenden Sie für die Rollenoption den ARN der Ausführungsrolle, die Sie in Schritt 2.2 erstellt haben. Geben Sie für die `vpc-config` durch Kommas getrennte Listen der Subnetze Ihrer Standard-VPC und die Sicherheitsgruppen-ID Ihrer Standard-VPC ein. Sie finden diese Werte in der Amazon-VPC-Konsole. Um die Subnetze Ihrer Standard-VPC zu finden, wählen Sie Ihre VPCs und dann die Standard-VPC Ihres AWS Kontos aus. Um die Sicherheitsgruppe für diese VPC zu finden, 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 AccessRedis \  
--region us-east-1 \  
--zip-file fileb://my_deployment_package.zip \  
--role arn:aws:iam::123456789012:role/elasticache-iam-auth-app \  
--handler app.lambda_handler \  
--runtime python3.12 \  
--timeout 30 \  
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-  
security-group-id
```

Schritt 3: Testen der Lambda -Funktion

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 (`AccessRedis`) mit dem Befehl `AWS Lambda invoke` auf.

```
aws lambda invoke \  
--function-name AccessRedis \  
--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/`). `AccessRedis` Die Ausgabe dieses Protokollstreams sollte ähnlich wie folgt aussehen:

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

- Überprüfen Sie die Ergebnisse in der `Konsole`. `AWS Lambda`

Schritt 4: Aufräumen (optional)

Gehen Sie zum Aufräumen wie folgt vor.

Themen

- [Schritt 4.1: Lambda-Funktion löschen](#)
- [Schritt 4.2: Löschen Sie den serverlosen Cache](#)
- [Schritt 4.3: Entfernen Sie die IAM-Rolle und -Richtlinien](#)

Schritt 4.1: Lambda-Funktion löschen

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

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: Entfernen Sie die IAM-Rolle und -Richtlinien

```
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"
```

```
--role-name "elasticache-iam-auth-app"
```

```
aws iam delete-policy \
```

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

Entwerfen und Verwalten eines eigenen ElastiCache-Clusters

Wenn Sie eine genaue Kontrolle über Ihren ElastiCache-Cluster benötigen, können Sie einen eigenen Cluster entwerfen. ElastiCache ermöglicht Ihnen den Betrieb eines knotenbasierten Clusters, indem Sie den Knotentyp, die Anzahl der Knoten und die Knotenplatzierung in den AWS Availability Zones für Ihren Cluster auswählen. Da ElastiCache ein vollständig verwalteter Service ist, werden Hardwarebereitstellung, Überwachung, Austausch von Knoten und Software-Patching für Ihren Cluster automatisch verwaltet.

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

Themen

- [ElastiCache für Redis-Komponenten und -Funktionen](#)
- [ElastiCache-for-Redis-Terminologie](#)
- [Entwerfen eines eigenen Clusters](#)
- [Knoten verwalten](#)
- [Verwalten von Clustern](#)
- [Vergleich der selbst entworfenen Caches von Memcached und Redis](#)
- [Online-Migration zu ElastiCache](#)
- [Auswahl von Regionen und Availability Zones](#)

ElastiCache für Redis-Komponenten und -Funktionen

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

Themen

- [ElastiCache Knoten](#)
- [ElastiCache für Redis-Shards](#)
- [ElastiCache für Redis-Cluster](#)

- [ElastiCache für die Redis-Replikation](#)
- [AWS Regionen und Verfügbarkeitszonen](#)
- [ElastiCache für Redis-Endpunkte](#)
- [ElastiCache Parametergruppen](#)
- [ElastiCache für die Redis-Sicherheit](#)
- [ElastiCache Subnetzgruppen](#)
- [ElastiCache für Redis-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 Teil eines sicheren, mit dem Netzwerk verbundenen RAMs mit fester Größe. 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 für Redis](#).

Jeder Knoten in einem Cluster ist vom selben Instance-Typ und führt dieselbe Cache-Engine aus. Jeder Cache-Knoten besitzt einen eigenen Domain Name Service (DNS)-Namen und Port. 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 Knotens 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 für ElastiCache](#).

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

ElastiCache für Redis-Shards

Ein Redis-Shard (in der API und CLI als Knotengruppe bezeichnet) ist eine Gruppierung von einem bis sechs zusammengehörigen Knoten. Ein Redis-Cluster (Clustermodus deaktiviert) 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.

Redis-Cluster (Cluster-Modus aktiviert) können bis zu 500 Shards haben, wobei Ihre Daten auf die Shards verteilt sind. Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Version der Redis-Engine 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. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht 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](#).

ElastiCache für Redis-Cluster

Ein Redis-Cluster ist eine logische Gruppierung einer oder mehrerer [ElastiCache für Redis-Shards](#). Die Daten werden über die Shards in einem Redis-Cluster (Cluster-Modus aktiviert) partitioniert.

Viele ElastiCache Operationen sind 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:

- [Verwalten von Clustern](#) und [Knoten verwalten](#)

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 von Clustern und Replikationsgruppen.

Typische Cluster-Konfigurationen

Folgende sind typische Clusterkonfigurationen.

Redis-Cluster

Redis-Cluster (Cluster-Modus deaktiviert) enthalten immer nur einen Shard (in der API und CLI eine Knotengruppe). Eine Redis-Shard enthält 1 bis 6 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.

Zur Verbesserung der Fehlertoleranz empfehlen wir, dass der Redis-Cluster mindestens zwei Knoten enthält und Multi-AZ aktiviert ist. Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Ändern sich die Anforderungen an Ihren Redis-Cluster (Cluster-Modus deaktiviert), können Sie ihn vergrößern oder verkleinern. Um dies zu tun, verschieben Sie den Cluster auf einen anderen Knoten-Instance-Typ. Sollte Ihre Anwendung sehr leseintensiv sein, empfehlen wir Ihnen, Redis-Cluster mit Nur-Lesereplikaten hinzuzufügen (Cluster-Modus deaktiviert). 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 unterstützt das dynamische Ändern des Knotentyps eines Redis-Clusters (Cluster-Modus deaktiviert) in einen größeren Knotentyp. Information zur Skalierung nach oben oder unten finden Sie unter [Skalieren von Clustern mit einem Knoten für Redis \(Clustermodus deaktiviert\)](#) oder [Skalieren von Redis-Clustern \(Clustermodus deaktiviert\) mit Replikatknoten](#).

ElastiCache für die Redis-Replikation

Die Replikation wird durch die Gruppierung von zwei bis sechs Knoten in einem Shard (in der API und CLI als Knotengruppe bezeichnet) realisiert. Einer dieser Knoten ist der primäre Knoten für Lese- und Schreibvorgänge. Alle anderen Knoten sind schreibgeschützte Replikationsknoten.

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](#).

Redis-Cluster (Cluster-Modus deaktiviert) unterstützen einen Shard (in der API und CLI als Knotengruppe bezeichnet).

Die Replikation aus Perspektive von API und CLI verwendet eine andere Terminologie, um die Kompatibilität mit früheren Versionen zu gewährleisten. Die Ergebnisse sind jedoch dieselben. Die folgende Tabelle zeigt die API- und CLI-Bedingungen für das Ausführen von Replikationen.

Vergleich der Replikation: Redis (Cluster-Modus deaktiviert) und Redis (Cluster-Modus aktiviert)

Einen Vergleich der Funktionen von Redis-Replikationsgruppen (Cluster-Modus deaktiviert) und Redis-Replikationsgruppen (Cluster-Modus aktiviert) finden Sie in der folgenden Tabelle.

| | Redis (Cluster-Modus deaktiviert) | Redis (Cluster-Modus aktiviert) |
|---------------------------------------|-----------------------------------|---------------------------------|
| Shards (Knotengruppen) | 1 | 1–500 |
| 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 die Sicherung und Wiederherstellung verwenden, um zu Redis zu migrieren (Cluster-Modus aktiviert).

Mithilfe von Sicherung und Wiederherstellung können Sie die Größe Ihres Redis-Clusters (Cluster-Modus aktiviert) ändern.

Sämtliche Shards (in der API und CLI, Knotengruppen) und Knoten müssen sich in derselben AWS -Region befinden. 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 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 ElastiCache für Redis erstellen, um Lesevorgänge mit geringer Latenz und regionsübergreifende Disaster Recovery zu ermöglichen. AWS Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#).

Replikation: Einschränkungen und Ausnahmen

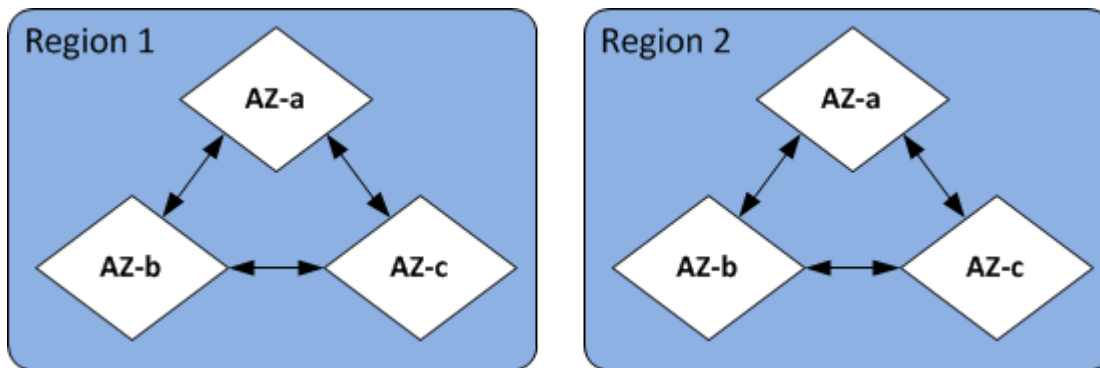
- Multi-AZ wird für den Knotentypen T1 nicht unterstützt.

AWS Regionen und Verfügbarkeitszonen

Amazon ElastiCache ist in mehreren AWS Regionen auf der ganzen Welt verfügbar. So können Sie ElastiCache Cluster an den Standorten einrichten, die Ihren Geschäftsanforderungen entsprechen. Sie können beispielsweise in der AWS Region starten, die Ihren Kunden am nächsten ist, oder um bestimmte gesetzliche Anforderungen zu erfüllen.

Standardmäßig verweisen die AWS SDKs AWS CLI, die ElastiCache API und die ElastiCache Konsole auf die Region USA West (Oregon). Im Zuge der ElastiCache Ausweitung der Verfügbarkeit auf neue AWS Regionen sind auch neue Endpunkte für diese AWS Regionen verfügbar. Sie können diese in Ihren HTTP-Anfragen, den AWS SDKs und ElastiCache der Konsole AWS CLI verwenden.

Jede AWS Region ist so konzipiert, dass sie vollständig von den anderen AWS Regionen isoliert ist. Innerhalb jeder gibt es mehrere Availability Zones (Verfügbarkeitszonen). Durch das Starten Ihrer Knoten in verschiedenen Availability Zones können Sie eine größtmögliche Fehlertoleranz zu erreichen. Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Availability Zones](#). In der folgenden Abbildung sehen Sie einen allgemeinen Überblick über die Funktionsweise von AWS Regionen und Availability Zones.



Informationen zu den von unterstützten AWS Regionen ElastiCache und ihren Endpunkten finden Sie unter [Unterstützte Regionen und Endpunkte](#).

ElastiCache für Redis-Endpunkte

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

Einzelknoten-Endpunkte für Redis (Cluster-Modus deaktiviert)

Der Endpunkt eines Redis-Clusters mit einem Knoten dient der Verbindung zum Cluster für Lese- und Schreibvorgänge.

Endpunkte mit mehreren Knoten für Redis (Cluster-Modus deaktiviert)

Ein Redis-Cluster mit mehreren Knoten (Cluster-Modus deaktiviert) 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-Endpunkt, um am Endpunkt ankommende Verbindungen auf alle Lesereplikate zu verteilen. Verwenden Sie die einzelnen Knotenendpunkte für Lesevorgänge (in der API/CLI werden diese als Leseendpunkte bezeichnet).

Redis-Endpunkte (Cluster Mode Enabled)

Ein Redis-Cluster (Cluster-Modus 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.

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

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

Ausführlichere Informationen zu ElastiCache Parametergruppen finden Sie unter [Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen](#)

ElastiCache für die Redis-Sicherheit

Aus Sicherheitsgründen ist der Knotenzugriff bei Redis auf Anwendungen beschränkt, ElastiCache die auf den Amazon EC2 EC2-Instances ausgeführt werden, die Sie zulassen. Sie können die Amazon-EC2-Instances, die auf Ihren Cluster zugreifen können, mithilfe von Sicherheitsgruppen kontrollieren.

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

For Redis beschränkt nicht nur den Zugriff auf Knoten, sondern unterstützt auch TLS und In-Place-Verschlüsselung ElastiCache für Knoten, auf denen bestimmte Versionen von for Redis ausgeführt werden. ElastiCache Weitere Informationen finden Sie hier:

- [Datensicherheit in Amazon ElastiCache](#)
- [Authentifizieren mit dem Redis-AUTH-Befehl](#)

ElastiCache Subnetzgruppen

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

Wird der Cluster in einer Amazon VPC erstellt, 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 Verwendung von Cache-Subnetzgruppen in einer Amazon-VPC-Umgebung finden Sie im Folgenden:

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Schritt 3: Zugriff auf den Cluster autorisieren](#)
- [Subnetze und Subnetzgruppen](#)

ElastiCache für Redis-Backups

Ein Backup ist eine point-in-time Kopie eines Redis-Clusters. 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 Redis-Version, die auf dem Cluster ausgeführt wird, benötigt der Sicherungsprozess für einen erfolgreichen Abschluss unterschiedlich große, reservierte Speichermengen. 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 Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#)

ElastiCache Ereignisse

ElastiCache Sendet bei wichtigen Ereignissen in einem Cache-Cluster eine Benachrichtigung an ein bestimmtes Amazon SNS SNS-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 [Amazon-SNS-Überwachung von ElastiCache-Ereignissen](#).

ElastiCache-for-Redis-Terminologie

Im Oktober 2016 hat Amazon ElastiCache den Support für Redis 3.2 eingeführt. Zu diesem Zeitpunkt haben wir der Support für die Partitionierung Ihrer Daten auf bis zu 500 Shards (in der ElastiCache-API und AWS CLI als Knotengruppen bezeichnet) hinzugefügt. Zur Aufrechterhaltung der Kompatibilität mit älteren Versionen erweiterten wir die Operationen der API-Version 2015-02-02 um die neue Redis-Funktionalität.

Gleichzeitig haben wir damit begonnen, in der ElastiCache-Konsole eine Terminologie zu verwenden, die für diese neue Funktionalität verwendet wird und in der Branche üblich ist. Durch diese Änderung kann es stellenweise dazu kommen, dass sich die in der API und der CLI verwendete Terminologie von der in der Konsole verwendeten unterscheidet. Die folgende Liste enthält Begriffe, die sich zwischen API, CLI und der Konsole unterscheiden können.

Cache-Cluster oder Knoten vs. Knoten

Es besteht eine Eins-zu-Eins-Beziehung zwischen einem Knoten und einem Cache-Cluster, wenn es keine Replikationsknoten gibt. In der ElastiCache-Konsole werden die Begriffe daher oft 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.

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

Cluster vs. Replikationsgruppe

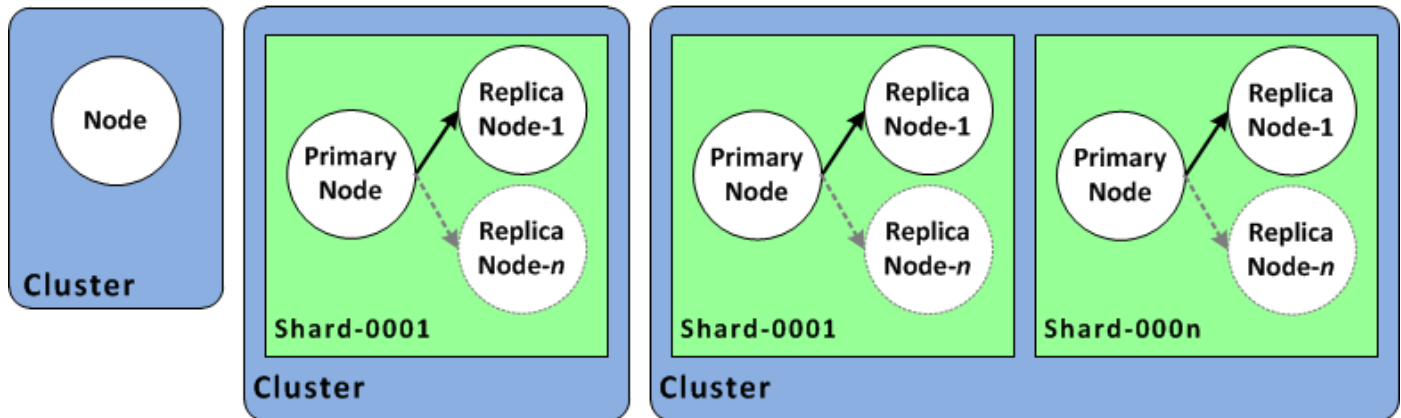
Die Konsole verwendet nun den Begriff Cluster für alle ElastiCache-for-Redis-Cluster. Die Konsole verwendet den Begriff "Cluster" in den folgenden Fällen:

- Wenn der Cluster ein Redis-Cluster mit einem Knoten ist
- Wenn es sich bei dem Cluster um einen Redis-Cluster (Cluster-Modus deaktiviert) handelt, der die Replikation innerhalb eines einzelnen Shards (in der API und CLI als Knotengruppe bezeichnet) unterstützt.
- Wenn es sich bei dem Cluster um einen Redis-Cluster (Cluster-Modus aktiviert) handelt, der die Replikation innerhalb von 1-90 Shards (oder bis zu 500 mit einer Anfrage 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 Replikationsgruppen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

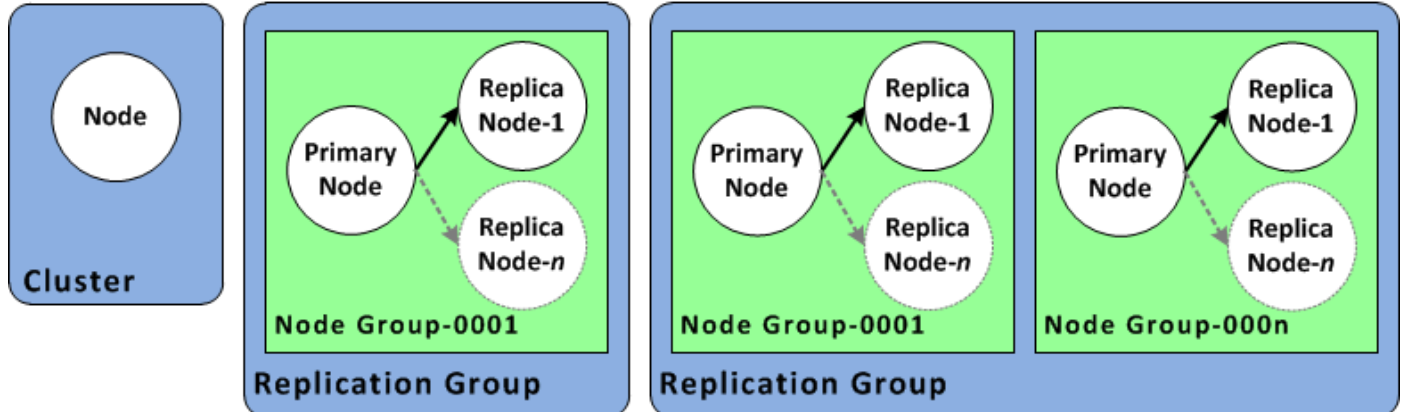
Das folgende Diagramm veranschaulicht die verschiedenen Topologien von ElastiCache für Redis-Clustern aus der Perspektive der Konsole.

ElastiCache für Redis: Console View



Die ElastiCache-API und AWS CLI-Vorgänge unterscheiden weiterhin zwischen ElastiCache-for-Redis-Clustern mit einem Knoten und Replikationsgruppen mit mehreren Knoten. Das folgende Diagramm veranschaulicht die verschiedenen ElastiCache-for-Redis-Topologien aus Sicht der ElastiCache-API und der AWS CLI-Perspektive.

ElastiCache für Redis: API/CLI View



Replikationsgruppe vs. globaler Datenspeicher

Ein globaler Datenspeicher ist eine Sammlung von einem oder mehreren Clustern, die über Regionen hinweg miteinander replizieren, während eine Replikationsgruppe Daten über einen Cluster-Modus-fähigen Cluster 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 AWS-Region befinden als der primäre Cluster.

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

Entwerfen eines eigenen Clusters

Im Folgenden finden Sie die einmaligen Aktionen, die Sie ergreifen müssen, um mit dem Entwerfen Ihres ElastiCache Clusters zu beginnen.

Themen

- [Einrichtung](#)
- [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 5: Löschen eines Clusters](#)
- [ElastiCache-Tutorials und -Videos](#)
- [Wie geht es weiter?](#)

Einrichtung

Erstellen Sie eine Subnetz-Gruppe, bevor Sie einen Cluster erstellen. Eine Cache-Subnetzgruppe ist eine Zusammenstellung von Subnetzen, die Sie vielleicht für Ihre Cache-Cluster in einer VPC festlegen möchten. Wenn Sie einen Cache-Cluster in einer VPC starten, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache verwendet dann diese Cache-Subnetzgruppe, um den einzelnen Cache-Knoten im Cluster IP-Adressen innerhalb dieses Subnetzes zuzuordnen.

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 von ElastiCache finden Sie unter [Einrichtung](#).

Schritt 1: Erstellung einer Subnetzgruppe

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-Managementkonsole 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 VPC ID die erstellte Amazon VPC aus.
 - 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 [↗](#)

ⓘ 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 Subnetzgruppen der ElastiCache-Konsole angezeigt. 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 AWS CLI-Thema [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 [Vorbereitung eines Clusters](#)-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 standardmäßigen - ElastiCache Nutzungsgebühren 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 ElastiCache Nutzungsgebühren finden Sie unter [Amazon ElastiCache](#).

Ihr Cluster wird, basierend auf dem Amazon-VPC-Service, in einer Virtual Private Cloud (VPC) gestartet.

Einen Redis-Cluster erstellen (Cluster-Modus deaktiviert) (Konsole)


So erstellen Sie einen Redis-Cluster (Cluster-Modus deaktiviert) mithilfe der ElastiCache Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon- 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 diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Create VPC (VPC erstellen) aus und befolgen Sie die Schritte unter [Erstellen einer Virtual Private Cloud \(VPC\)](#).
5. Wählen Sie auf der ElastiCache Dashboard-Seite Redis-Cache und dann Redis-Cache erstellen 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) 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 [Minimieren von Ausfallzeiten in ElastiCache für Redis 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 [Redis-spezifische Parameter](#) und [Erstellen einer Parametergruppe](#).

 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 (▼) 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](#).

- 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,

dem sowohl IPv4- als auch IPv6-Adressen zugewiesen sind, um im Dual-Stack-Modus zu arbeiten, und ein IPv6-onlySubnetz, um als IPv6-onlySubnetz zu arbeiten.

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

Weitere Informationen finden Sie hier:

- [Auswählen eines Netzwerktyps](#).
- [Erstellen eines Subnetzes in der VPC](#)

Wenn Sie [Verwenden von Local Zones 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 aus.
- 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 Availability Zones](#).

5. Wählen Sie Next (Weiter).

6. Unter Advanced Redis settings (Erweiterte Redis-Einstellungen):

- 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 bereitzustellen, indem Sie Kundenverwalteter AWS KMS-Schlüssel und den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwendung von kundenverwalteten Schlüsseln aus 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 bei Redis-Engine Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Optionen für die Zugriffskontrolle 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 [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
 - Redis AUTH Standardbenutzer - Ein Authentifizierungsmechanismus für den Redis-Server. Weitere Informationen finden Sie unter [Redis AUTH](#).
- Redis AUTH – Ein Authentifizierungsmechanismus für Redis-Server. Weitere Informationen finden Sie unter [Redis AUTH](#).

Note

Für Redis-Versionen ab 3.2.6, ausgenommen Version 3.2.10, ist Redis AUTH 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 Ihre VPC](#) im Benutzerhandbuch zu Amazon VPC.

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 zu Redis-Sicherungen und -Wiederherstellungen finden Sie unter [Snapshot und Wiederherstellung](#).

8. (Optional) Geben Sie ein Wartungsfenster an. Das Maintenance window (Wartungsfenster) ist der (in der Regel eine Stunde lange) allwöchentliche Zeitraum, für den ElastiCache die Systemwartung Ihres Clusters plant. Sie können es ElastiCache überlassen, einen Tag und eine Uhrzeit für das Wartungsfenster auszuwählen (No preference (Keine Präferenz)) oder Sie können den Tag, die Uhrzeit und die Dauer selbst wählen (Specify maintenance window (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 Uhrzeiten sind in UCT angegeben.

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

9. (Optional) Für Protokolle:
 - Unter Protokollformat wählen Sie entweder Text oder JSON aus.
 - Wählen Sie unter Zieltyp entweder CloudWatch Protokolle oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen und geben Sie entweder Ihren CloudWatch Protokollgruppennamen oder Ihren Firehose-Streamnamen ein oder wählen Sie Vorhanden auswählen und wählen Sie dann entweder Ihren CloudWatch Protokollgruppennamen oder Ihren Firehose-Streamnamen aus.
10. Um Sie bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen zu unterstützen, können Sie für Tags jeder Ressource eigene Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Markieren Ihrer ElastiCache-Ressourcen](#).
11. Wählen Sie Weiter aus.
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 [Minimieren von Ausfallzeiten in ElastiCache für Redis 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, sich mit ihm verbinden und ihn nutzen. Weitere Informationen finden Sie unter [Schritt 3: Zugriff auf den Cluster autorisieren](#) und [Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten](#).

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](#).

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

Example

Der folgende CLI-Code erstellt einen Redis-Cache-Cluster (Cluster-Modus 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 Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#).
- Informationen zur Verwendung der AWS CLI finden Sie unter [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(AWS CLI\)](#).

Schritt 3: Zugriff auf den Cluster autorisieren

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

Alle ElastiCache-Cluster sind für den Zugriff über eine Amazon-EC2-Instance konzipiert. Das häufigste Szenario ist der Zugriff auf einen ElastiCache-Cluster von einer Amazon-EC2-Instance in derselben Amazon Virtual Private Cloud (Amazon VPC), was in 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 eine Verbindung zu einem Cluster herstellen können, müssen Sie der EC2-Instance Zugriff auf den Cluster gewähren. Die erforderlichen Schritte sind davon abhängig, ob der Cluster in EC2-VPC oder EC2-Classic gestartet wurde.

Der gängigste Anwendungsfall ist, wenn eine auf einer EC2-Instance bereitgestellte Anwendung eine Verbindung mit einem Cluster in der gleichen VPC herstellen muss. Nachfolgend finden Sie den leichtesten Weg für die Verwaltung des Zugriffs zwischen EC2-Instances und DB-Instances in derselben VPC:

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 für diese Sicherheitsgruppe beispielsweise eine benutzerdefinierte Regel erstellen, die TCP-Zugriff über den Port, den Sie dem Cluster bei seiner Erstellung zugeordnet haben, und eine IP-Adresse gewährt, mit der Sie auf den Cluster zugreifen.

Der Standard-Port für Redis-Cluster und -Replikationsgruppen ist 6379.

Important

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

Wenn Sie ElastiCache-Knoten in einer Amazon-VPC ausführen, steuern Sie den Zugriff auf Ihre Cluster über Amazon-VPC-Sicherheitsgruppen, die sich von ElastiCache-Sicherheitsgruppen unterscheiden. Weitere Informationen zum Verwenden von

ElastiCache in einer Amazon VPC, finden Sie unter [Amazon VPCs und ElastiCache Sicherheit](#).

2. Erstellen Sie eine VPC-Sicherheitsgruppe für Ihre EC2-Instances (Web- und Anwendungsserver). Mithilfe dieser Sicherheitsgruppe können Sie bei Bedarf den Internetzugriff auf die EC2-Instance über die Routing-Tabelle der VPC zulassen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, damit der TCP-Zugriff auf die EC2-Instance über Port 22 möglich ist.
3. Erstellen Sie benutzerdefinierte Regeln in der Sicherheitsgruppe für Ihren Cluster, die Verbindungen über die von Ihnen für EC2-Instances erstellte Sicherheitsgruppe zulassen. Damit wird jedem Mitglied der Sicherheitsgruppe der Zugriff auf die DB-Instances gestattet.

Note

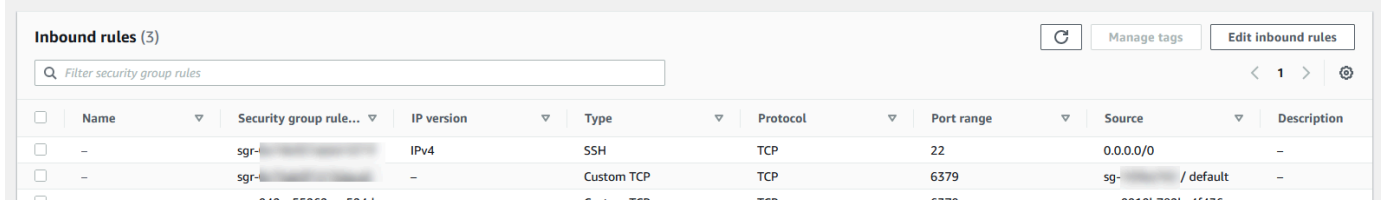
Wenn Sie planen, [Local Zones](#) zu nutzen, stellen Sie sicher, dass Sie sie aktiviert haben. Wenn Sie eine Subnetzgruppe in dieser Local Zone erstellen, wird Ihre VPC auf diese Local Zone erweitert, und Ihre VPC behandelt das Subnetz wie jedes Subnetz in einer anderen Availability Zone. Alle relevanten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

So erstellen Sie eine Regel in einer VPC-Sicherheitsgruppe, die Verbindungen über eine andere Sicherheitsgruppe zulässt

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc>.
2. Wählen Sie im Navigationsbereich Sicherheitsgruppen aus.
3. 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.
4. Wählen Sie für Type die Option Custom TCP Rule aus.
 - a. Geben Sie für Port Range den Port an, den Sie beim Erstellen des Clusters verwendet haben.

Der Standard-Port für 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 Amazon-EC2-Instances verwenden werden.
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 einer anderen Amazon VPC, einer anderen AWS-Region oder sogar von Ihrem Unternehmensnetzwerk finden Sie unter:

- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#)
- [Zugriff auf ElastiCache-Ressourcen außerhalb von 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 Amazon-EC2-Instance erstellt haben und eine Verbindung mit ihr möglich ist. Anweisungen dazu finden Sie im [Amazon-EC2-Erste-Schritte-Leitfaden](#).

Eine Amazon-EC2-Instance kann sich nur dann mit einem Cluster-Knoten verbinden, wenn Sie sie dazu autorisiert haben.

Finden Sie Ihre Knotenendpunkte

Sobald der Cluster den Status verfügbar hat und Sie autorisierten Zugriff haben, können Sie sich bei einer Amazon-EC2-Instance anmelden und eine Verbindung zum Cluster herstellen. Hierzu müssen Sie zuerst den Endpunkt bestimmen.

Suche nach den Endpunkten eines Redis-Clusters (Cluster-Modus deaktiviert) (Konsole)

Wenn ein Redis-Cluster (Cluster-Modus 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 auf den 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 wird eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Lesereplikate in einem ElastiCache-for-Redis-Cluster aufteilen. 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 Lese-Replikate Ihres ElastiCache for Redis-Clusters in verschiedenen AWS-Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Lese-Endpunkte zu gewährleisten.

Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS-Eintrag, der nach dem Rotationsprinzip zu einer IP-Adresse eines der Replikationsknoten 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 Redis-Clusters (Cluster-Modus deaktiviert)

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

Der Cluster-Bildschirm erscheint mit einer Liste von vorhandenen Redis-Serverless-Clustern, Redis-Clustern (Cluster-Modus deaktiviert) und Redis-Clustern (Cluster-Modus aktiviert). Wählen Sie den Cluster, den Sie im Abschnitt [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#) 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 | 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 und Leser-Endpunkte für einen Redis-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 Redis-Cluster (Cluster-Modus deaktiviert) über Replikationsknoten verfügt, können Sie die Replikationsknoten-Endpunkte des Clusters finden, indem Sie den Namen des Clusters und die Registerkarte Nodes (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 Redis-Cluster (Cluster-Modus 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 (Cluster-Modus 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 ermitteln](#)
- [Suchen von Endpunkten für einen Redis-Cluster \(Cluster-Modus aktiviert\) \(Konsole\)](#)—Sie benötigen den Konfigurationsendpunkt des Clusters.
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Finden von Endpunkten \(ElastiCache-API\)](#)

Verbinden Sie sich mit einem Redis-Cluster oder Replikationsgruppe (Linux)

Nachdem Sie den gewünschten Endpunkt gefunden haben, können Sie sich bei einer EC2-Instance anmelden und eine Verbindung zu dem Cluster oder der Replikationsgruppe herstellen. Im folgenden Beispiel verwenden Sie das `redis-cli`-Dienstprogramm, um eine Verbindung mit einem Cluster herzustellen. Die neueste Version von `redis-cli` unterstützt auch SSL/TLS für die Verbindung von verschlüsselungs-/authentifizierungsfähigen Clustern.

Im folgenden Beispiel werden Amazon-EC2-Instances verwendet, auf denen Amazon Linux und Amazon Linux 2 ausgeführt wird. Details zum Installieren und Kompilieren von `redis-cli` mit anderen Linux-Distributionen finden Sie in der Dokumentation für Ihr spezifisches Betriebssystem.

Note

Dieser Prozess umfasst das Testen einer Verbindung mit dem `redis-cli`-Dienstprogramm nur für ungeplante Verwendung. Eine Liste der unterstützten Redis-Clients finden Sie in der [Redis-Dokumentation](#). Beispiele für die Verwendung des AWS-SDKs mit ElastiCache finden Sie unter [Erste Schritte mit ElastiCache und AWS-SDKs](#).

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

1. Führen Sie den folgenden Befehl aus, um sich mit dem Cluster zu verbinden, und ersetzen Sie *Primärer Endpunkt* und *Portnummer* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Redis ist 6379.)

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

Es erscheint eine Redis-Eingabeaufforderung ähnlich wie die Folgende:

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

2. Sie können jetzt die Redis-Befehle 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 sich mit dem Cluster zu verbinden, und ersetzen Sie *Konfigurations-Endpunkt* und *Portnummer* mit dem Endpunkt Ihres Clusters und Ihrer Portnummer. (Der Standardport für Redis ist 6379.)

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

Note

Im vorherigen Befehl aktiviert Option -c den Cluster-Modus nach den [-ASK und -MOVED-Umleitungen](#).

Es erscheint eine Redis-Eingabeaufforderung ähnlich wie die Folgende:

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

2. Sie können jetzt die Redis-Befehle 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 die Fehlermeldung MOVED zurück. Weitere Informationen zum MOVED-Fehler finden Sie unter [Redis-Cluster-Spezifikation](#).

```
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 redis-cli eine unverschlüsselte TCP-Verbindung bei der Verbindung zu Redis. Die Option `BUILD_TLS=yes` aktiviert SSL/TLS zum Zeitpunkt der Redis-CLI-Kompilierung, wie im vorhergehenden [Herunterladen und Einrichten von redis-cli](#)-Abschnitt gezeigt. Die Aktivierung von AUTH ist optional. Sie müssen jedoch Verschlüsselung während der Übertragung aktivieren, um AUTH zu aktivieren. Weitere Informationen zur ElastiCache-Verschlüsselung und -Authentifizierung finden Sie unter [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#).

Note

Sie können die Option `--tls` mit redis-cli verwenden, um sich sowohl mit verschlüsselten Clustern mit aktiviertem als auch mit deaktiviertem Cluster-Modus zu verbinden. Wenn für einen Cluster ein AUTH-Token festgelegt ist, können Sie die Option `-a` verwenden, um ein AUTH-Passwort anzugeben.

Ersetzen Sie in den folgenden Beispielen den *Cluster-Endpunkt* und die *Portnummer* durch den Endpunkt Ihres Clusters und Ihre Portnummer. (Der Standardport für Redis ist 6379.)

Verbindung zu verschlüsselten Clustern mit Cluster-Modus deaktiviert

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

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

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

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

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

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

Nach der Verbindung mit dem Cluster können Sie die Redis-Befehle ausführen, wie in den vorangegangenen Beispielen für unverschlüsselte Cluster gezeigt.

Redis-CLI-Alternative

Wenn der Cluster nicht im Cluster-Modus aktiviert ist und Sie eine Verbindung zum Cluster für einen kurzen Test herstellen müssen, aber ohne die redis-cli-Kompilierung zu durchlaufen, können Sie telnet oder openssl verwenden. Ersetzen Sie in den folgenden Beispielbefehlen den *Cluster-Endpunkt* und die *Portnummer* durch den Endpunkt Ihres Clusters und Ihre Portnummer. (Der Standardport für Redis 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. Im folgenden Beispiel ersetzen Sie *Ihr Passwort* mit dem Passwort für Ihren Cluster zurück.

```
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
```

Verbindung mit einem Redis-Cluster oder einer Replikationsgruppe (Windows)

Um eine Verbindung von einer EC2 Windows-Instance über die Redis-CLI mit dem Redis-Cluster herzustellen, müssen Sie das redis-cli-Paket herunterladen und redis-cli.exe verwenden, um eine Verbindung mit dem Redis-Cluster von einer EC2 Windows-Instance herzustellen.

Im folgenden Beispiel verwenden Sie das redis-cli-Dienstprogramm, um eine Verbindung mit einem Cluster herzustellen, der nicht verschlüsselungsfähig ist und Redis ausführt. Weitere Informationen über Redis und verfügbare Redis-Befehle finden Sie unter [Redis-Befehle](#) auf der Redis-Webseite.

So richten Sie eine Verbindung mit einem nicht verschlüsselungsfähigen Redis-Cluster mit redis-cli ein

1. Verbinden Sie sich mit Ihrer Amazon-EC2-Instance mit dem Verbindungsdienstprogramm Ihrer Wahl. Anweisungen zur Verbindung mit einer Amazon-EC2-Instance finden Sie im [Amazon-EC2-Erste-Schritte-Handbuch](#).
2. Kopieren Sie den Link <https://github.com/microsoftarchive/redis/releases/download/win-3.0.504/Redis-x64-3.0.504.zip> und fügen Sie ihn in einen Internetbrowser ein, um die ZIP-Datei für den Redis-Client von der verfügbaren Version bei Github unter <https://github.com/microsoftarchive/redis/releases/tag/win-3.0.504> herunterzuladen

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

Öffnen Sie die Eingabeaufforderung, wechseln Sie zum Redis-Verzeichnis und führen Sie den Befehl `c:\Redis>redis-cli -h Redis_Cluster_Endpoint -p 6379` aus.

Zum Beispiel:

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

3. Ausführen von Redis-Befehlen

Sie sind jetzt mit dem Cluster verbunden und können 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 redis-cli
```

Schritt 5: 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-for-Redis-Cluster löschen, bleiben Ihre manuellen Snapshots erhalten. 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](#).

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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie Redis im Dashboard der ElastiCache-Konsole aus.

Eine Liste aller Caches, die Redis ausführen, wird eingeblendet.

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 immer nur einen Cluster auf einmal von der ElastiCache-Konsole aus 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.

Verwendung der 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 CLI-Aktion `delete-cache-cluster` 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 unter AWS CLI für das ElastiCache-Thema [delete-cache-cluster](#).

ElastiCache-Tutorials und -Videos

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

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

ElastiCache-Videos

Im Folgenden finden Sie Videos, die Ihnen helfen, grundlegende und fortgeschrittene Konzepte von Amazon ElastiCache zu erlernen. Informationen zum AWS-Training finden Sie unter [AWSTraining und Zertifizierung](#).

Themen

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

Einführungsvideos

The following videos introduce you to Amazon ElastiCache.

Themen

- [AWS re:Invent 2020: Was gibt es Neues bei Amazon ElastiCache?](#)
- [AWS re:Invent 2019: Was gibt es Neues bei Amazon ElastiCache?](#)
- [AWS re:Invent 2017: Was gibt es Neues bei Amazon ElastiCache?](#)
- [DAT204—Erstellen von skalierbaren Anwendungen auf AWS-NoSQL-Diensten \(re:Invent 2015\)](#)
- [DAT207—Beschleunigung der Anwendungsleistung mit Amazon ElastiCache \(AWS re:Invent 2013\)](#)

[AWS re:Invent 2020: Was gibt es Neues bei Amazon ElastiCache?](#)

[AWS re:Invent 2020: Was gibt es Neues bei Amazon ElastiCache?](#)

[AWS re:Invent 2019: Was gibt es Neues bei Amazon ElastiCache?](#)

[AWS re:Invent 2019: Was gibt es Neues bei Amazon ElastiCache?](#)

[AWS re:Invent 2017: Was gibt es Neues bei Amazon ElastiCache?](#)

[AWS re:Invent 2017: Was gibt es Neues bei Amazon ElastiCache?](#)

[DAT204—Erstellen von skalierbaren Anwendungen auf AWS-NoSQL-Diensten \(re:Invent 2015\)](#)

In dieser Sitzung werden die Vorteile von NoSQL-Datenbanken erörtert und die wichtigsten von AWS—Amazon-DynamoDB und Amazon ElastiCache angebotenen NoSQL-Dienste vorgestellt. Anschließend berichten zwei führende Kunden, Expedia und Mapbox, über ihre Anwendungsfälle

und architektonischen Herausforderungen und wie sie diese mit Hilfe von AWS-NoSQL-Diensten, einschließlich Design Patterns und Best Practices, gelöst haben. Am Ende der Sitzung werden Sie NoSQL und die leistungsstarken Fähigkeiten besser verstehen. Sie werden sich den Herausforderungen, die eine Datenbank darstellen kann, zuversichtlich gewachsen fühlen.

[DAT204—Erstellen von skalierbaren Anwendungen auf AWS-NoSQL-Diensten \(re:Invent 2015\)](#)

DAT207—Beschleunigung der Anwendungsleistung mit Amazon ElastiCache (AWS re:Invent 2013)

In diesem Video erfahren Sie, wie Sie mit Amazon ElastiCache ganz einfach ein In-Memory-Caching-System bereitstellen können, um die Leistung Ihrer Anwendung zu steigern. Wir zeigen Ihnen, wie Sie Amazon ElastiCache nutzen können, um die Latenzzeit Ihrer Anwendung 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 des Caching profitieren können, und diskutieren die von Amazon ElastiCache bereitgestellten Funktionen.

[DAT207 – Beschleunigung der Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)](#)

Weiterführende Videos

Die folgenden Videos behandeln fortgeschrittenere Themen zu Amazon ElastiCache.

Themen

- [Bewährte Methoden für erfolgreiches Design in Amazon ElastiCache \(re:Invent 2020\)](#)
- [Echtzeitanwendungen mit Amazon ElastiCache optimieren \(re:Invent 2019\)](#)
- [Bewährte Methoden: Migration von Redis-Clustern von Amazon EC2 zu ElastiCache \(re:Invent 2019\)](#)
- [Skalierung einer Fantasy-Sports-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Zuverlässiges und skalierbares Redis in der Cloud mit Amazon ElastiCache \(re:Invent 2018\)](#)
- [Vertiefter Einblick in ElastiCache: Entwurfsmuster für In-Memory-Datenspeicher \(re:Invent 2018\)](#)
- [DAT305—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2017\)](#)
- [DAT306—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2016\)](#)
- [DAT317—Wie IFTTT ElastiCache for Redis zur Vorhersage von Ereignissen verwendet \(re:Invent 2016\)](#)
- [DAT407—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2015\)](#)

- [SDD402—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2014\)](#)
- [DAT307—Vertiefter Einblick in Architektur und Entwurfsmuster von Amazon ElastiCache \(re:Invent 2013\)](#)

Bewährte Methoden für erfolgreiches Design in Amazon ElastiCache (re:Invent 2020)

Mit dem explosionsartigen Wachstum von geschäftskritischen Echtzeitanwendungen, die auf Redis basieren, sind Verfügbarkeit, Skalierbarkeit und Sicherheit zu den wichtigsten Faktoren geworden. Lernen Sie bewährte Methoden für die Einrichtung von Amazon ElastiCache kennen, damit Sie mit Online-Skalierung, hoher Verfügbarkeit in Multi-AZ-Bereitstellungen und Sicherheitskonfigurationen erfolgreich sind.

[Bewährte Methoden für erfolgreiches Design in Amazon ElastiCache \(re:Invent 2020\)](#)

Echtzeitanwendungen mit Amazon ElastiCache optimieren (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 den vollständig verwalteten, In-Memory-Amaon-ElastiCache für extreme Leistung, hohe Skalierbarkeit, Verfügbarkeit und Sicherheit nutzen.

[Echtzeitanwendungen mit Amazon ElastiCache optimieren \(re:Invent 2019:\)](#)

Bewährte Methoden: Migration von Redis-Clustern von Amazon EC2 zu ElastiCache (re:Invent 2019)

Die Verwaltung von Redis-Clustern in Eigenregie kann schwierig sein. 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 einfach von selbst gehostetem Redis auf Amazon EC2 zu vollständig verwaltetem Amazon ElastiCache verschieben, während der Cluster-Modus deaktiviert ist. In dieser Einheit erfahren Sie mehr über das neue Online-Migrationstool, sehen eine Demo und lernen vor allem praktische bewährte Methoden für eine reibungslose Migration zu Amazon ElastiCache.

[Bewährte Methoden: Migration von Redis-Clustern von Amazon EC2 zu ElastiCache \(re:Invent 2019\)](#)

Skalierung einer Fantasy-Sports-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 Beitrag erklärt Amit Sharma, CTO von Dream11, wie das Unternehmen Amazon Aurora und Amazon ElastiCache einsetzt, um den Flash-Datenverkehr zu bewältigen, der sich innerhalb eines 30-Sekunden-Antwortfensters 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 Sports-Plattform mit Amazon ElastiCache und Amazon Aurora (STP11)

[Skalierung einer Fantasy-Sports-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 \(re:Invent 2018\)](#)

Zuverlässiges und skalierbares Redis in der Cloud mit Amazon ElastiCache (re:Invent 2018)

Diese Sitzung behandelt die Funktionen und Verbesserungen in unserem Redis-kompatiblen Service, Amazon ElastiCache for Redis. Wir behandeln die wichtigsten Funktionen, wie Redis 5, Skalierbarkeit und Leistungsverbesserungen, Sicherheit und Compliance und vieles mehr. Wir besprechen auch zukünftige Funktionen und Kundenfallstudien.

[Zuverlässiges und skalierbares Redis in der Cloud mit Amazon ElastiCache \(re:Invent 2018\)](#)

Vertiefter Einblick in ElastiCache: Entwurfsmuster für In-Memory-Datenspeicher (re:Invent 2018)

In dieser Sitzung werfen wir einen Blick hinter die Kulissen und erfahren mehr über das Design und die Architektur von Amazon ElastiCache. Informieren Sie sich über gemeinsame Designmuster mit unseren Redis- und Memcached-Angeboten und darüber, wie Kunden diese für die speicherinterne Datenverarbeitung nutzen, um die Latenzzeit zu verringern und den Anwendungsdurchsatz zu verbessern. Wir überprüfen bewährte Methoden für ElastiCache, Entwurfsmuster und Gegenbeispiele.

[Vertiefter Einblick in ElastiCache: Entwurfsmuster für In-Memory-Datenspeicher \(re:Invent 2018\)](#)

DAT305—Vertiefter Einblick in Amazon ElastiCache (re:Invent 2017)

Schauen Sie hinter die Kulissen und erfahren Sie mehr über das Design und die Architektur von Amazon ElastiCache. Unsere Angebote für Memcached und Redis enthalten gängige Entwurfsmuster

sowie Informationen dazu, wie Kunden diese für In-Memory-Operationen zur Erhöhung von Latenz und Durchsatz der Anwendungen verwendet haben. In diesem Video werden bewährte ElastiCache-Verfahren, Entwurfsmuster und Gegenbeispiele vorgestellt.

Das Video zeigt Folgendes:

- ElastiCache for Redis Online-Resharding
- ElastiCache Sicherheit und Verschlüsselung
- ElastiCache-for-Redis-Version 3.2.10

[DAT305—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2017\)](#)

DAT306—Vertiefter Einblick in Amazon ElastiCache (re:Invent 2016)

Schauen Sie hinter die Kulissen und erfahren Sie mehr über das Design und die Architektur von Amazon ElastiCache. Unsere Angebote für Memcached und Redis enthalten gängige Entwurfsmuster sowie Informationen dazu, wie Kunden diese für In-Memory-Operationen zur Erhöhung von Latenz und Durchsatz der Anwendungen verwendet haben. In dieser Sitzung werden bewährte ElastiCache-Verfahren, Entwurfsmuster und Gegenbeispiele vorgestellt.

[DAT306—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2016\)](#)

DAT317—Wie IFTTT ElastiCache for Redis zur Vorhersage von Ereignissen verwendet (re:Invent 2016)

IFTTT ist ein kostenloser Service, mit dem Kunden die Services, die sie am liebsten verwenden, noch besser nutzen können. Das reicht von der Automatisierung einfacher Aufgaben bis hin zur Interaktion mit und Steuerung von Wohnungen und Häusern. IFTTT verwendet ElastiCache for Redis, um den Transaktionsverlauf und Zeitplanvorhersagen sowie Indizes für Protokolldokumente auf Amazon S3 zu speichern. In dieser Sitzung erfahren Sie, wie Kunden mit der Skriptleistung von Lua und den Datentypen von Redis etwas schaffen können, das anderswo unerreichbar gewesen wäre.

[DAT317—Wie IFTTT ElastiCache for Redis zur Vorhersage von Ereignissen verwendet \(re:Invent 2016\)](#)

DAT407—Vertiefter Einblick in Amazon ElastiCache (re:Invent 2015)

Werfen Sie einen Blick hinter die Kulissen und erfahren Sie mehr über das Design und die Architektur von Amazon ElastiCache. Sehen Sie sich die gängigen Entwurfsmuster in unseren Angeboten für Redis und Memcached an und erfahren Sie, wie Kunden diese für In-Memory-Operationen zur

Verbesserung von Latenz und Durchsatz der Anwendungen verwendet haben. In dieser Sitzung werden bewährte Methoden, Design Patterns und Gegenbeispiele im Zusammenhang mit Amazon ElastiCache besprochen.

[DAT407—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2015\)](#)

SDD402—Vertiefter Einblick in Amazon ElastiCache (re:Invent 2014)

In diesem Video untersuchen wir gängige Caching-Anwendungsfälle, die Memcached- und Redis-Engines, Muster für die Auswahl der besten Engine für Ihre Anforderungen, konsistentes Hashing und mehr zur Erstellung schneller, skalierbarer Anwendungen. Frank Wiebe, Principal Scientist bei Adobe, erläutert, wie Adobe Amazon ElastiCache einsetzt, um das Kundenerlebnis zu verbessern und das Geschäft zu skalieren.

[DAT402—Vertiefter Einblick in Amazon ElastiCache \(re:Invent 2014\)](#)

DAT307—Vertiefter Einblick in Architektur und Entwurfsmuster von Amazon ElastiCache (re:Invent 2013)

In diesem Video untersuchen wir Caching, Caching-Strategien, Skalierung und Überwachung. Wir vergleichen auch die Memcached- und Redis-Engines. Während dieser Einheit werden auch bewährte Methoden und Entwurfsmuster im Zusammenhang mit Amazon ElastiCache besprochen.

[DAT307 – Vertiefter Einblick in Architektur und Entwurfsmuster von Amazon ElastiCache \(AWS re:Invent 2013\)](#)

Wie geht es weiter?

Nachdem Sie die Übung „Erste Schritte“ ausprobiert haben, können Sie die folgenden Abschnitte für weitere Informationen über ElastiCache und die verfügbaren Tools durchsuchen:

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

Nachdem Sie die Übung „Erste Schritte“ abgeschlossen haben, können Sie die folgenden Abschnitte für weitere Informationen über die Verwaltung von ElastiCache durchlesen:

- [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 Best Practices und Caching-Strategien](#)

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

Knoten verwalten

Ein Knoten ist der kleinste Baustein einer Amazon-ElastiCache-Bereitstellung. Er ist ein RAM-Bereich fester Größe, der sicher ins Netzwerk eingebunden ist. Auf jedem Knoten wird die Engine ausgeführt, je nachdem, welche Option beim Erstellen oder der letzten Änderung des Clusters oder der Replikationsgruppe ausgewählt wurde. Jeder Knoten besitzt einen eigenen Domain Name Service (DNS)-Namen und Port. Es werden mehrere Arten von ElastiCache-Knoten unterstützt, die jeweils über unterschiedliche Mengen an zugehörigem Speicher und Rechenleistung verfügen.

Im Allgemeinen haben Redis-Bereitstellungen (mit aktiviertem Cluster-Modus), aufgrund des Supports für Sharding, eine Reihe kleinerer Knoten. Im Gegensatz dazu haben Redis-Implementierungen (Cluster-Modus deaktiviert) weniger und größere Knoten in einem Cluster. Eine ausführliche Erörterung der zu verwendenden Knotengröße finden Sie unter [Auswahl der Knotengröße](#).

Themen

- [ElastiCache Knotenstatus anzeigen](#)
- [Redis-Knoten und -Shards](#)
- [Mit Knoten verbinden](#)
- [Unterstützte Knotentypen](#)
- [Neustart von Knoten \(nur mit deaktiviertem Cluster-Modus\)](#)
- [Ersetzen von Knoten](#)
- [Reservierte Knoten für ElastiCache](#)
- [Migration von Knoten der vorherigen Generation](#)

Einige wichtige Operationen, die Knoten umfassen, sind folgende:

- [Knoten zu einem Cluster hinzufügen](#)

- [Entfernen von Knoten aus einem Cluster](#)
- [Skalierung ElastiCache für Redis](#)
- [Verbindungsendpunkte ermitteln](#)

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 | Ein Aktualisierungsstatus gibt an, dass eine oder mehrere der folgenden Bedingungen für den ElastiCache Amazon-Knoten zutreffen: |

| Typ | Berechnet | Beschreibung |
|-----|-----------|---|
| | | <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 ElastiCache Cluster aktualisiert.• Der ElastiCache Cluster wird hoch- oder herunterkaliert.• Die Konfigurationen für die Protokollzustellung werden für den ElastiCache Cluster geändert.• Ein Löschvorgang für den ElastiCache Knoten steht noch aus.• Das Passwort ElastiCache für Redis wird aktualisiert/rotiert mit AWS Secrets Manager |

| Typ | Berechnet | Beschreibung |
|--|-----------------|--|
| <code>rebooting cache cluster nodes</code> | Berechnet | Der ElastiCache Knoten wird aufgrund einer Kundenanfrage oder eines ElastiCache Amazon-Prozesses, der einen Neustart des Knotens erfordert, neu gestartet. |
| <code>incompatible_parameters</code> | Nicht berechnet | 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. |

| Typ | Berechnet | Beschreibung |
|-----------------------------------|-----------------|---|
| <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 (Amazon VPC) 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 ElastiCache for 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 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

Verwenden Sie den `describe-cache-cluster` Befehl, um den ElastiCache Knoten und seine Statusinformationen mit dem AWS CLI anzuzeigen. Mit dem folgenden AWS CLI Befehl werden beispielsweise alle ElastiCache Knoten angezeigt.

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

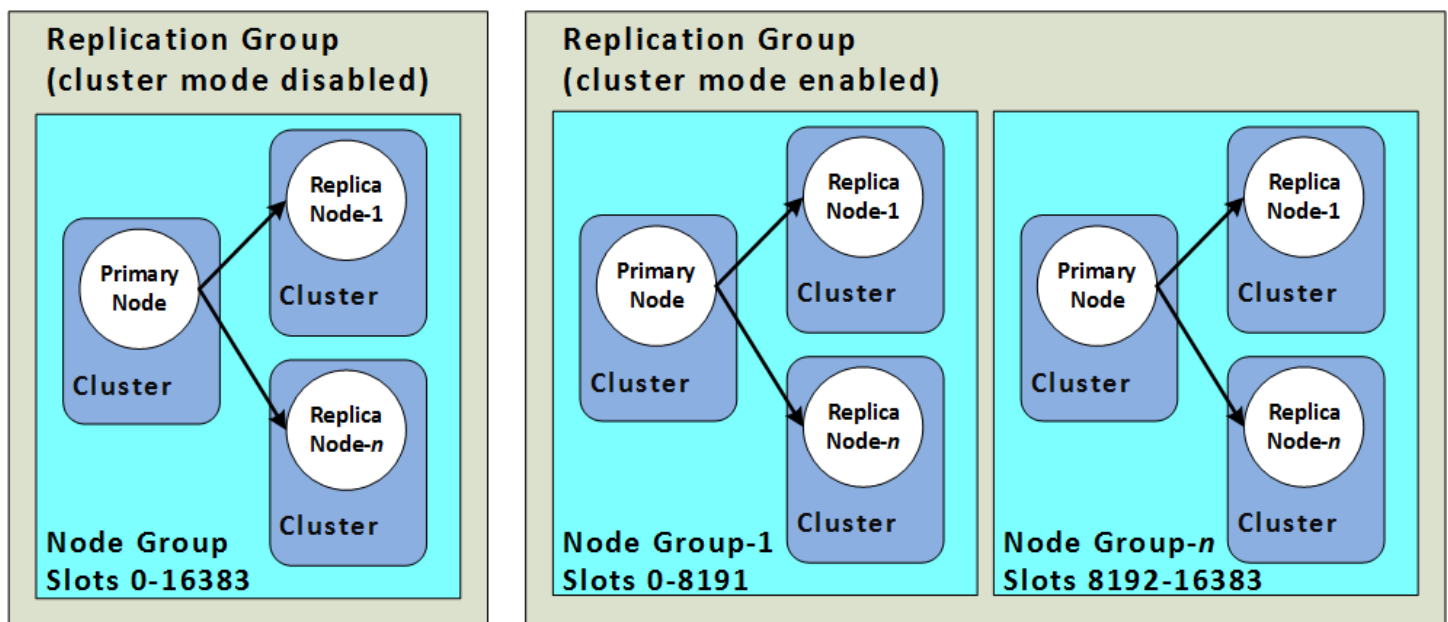
Den ElastiCache Knotenstatus über die API anzeigen

Um den Status des ElastiCache Nodes mithilfe der ElastiCache Amazon-API einzusehen, rufen Sie die `DescribeCacheClusterOperation` mit dem `ShowCacheNodeInfo` Flag auf, um Informationen über die einzelnen Cache-Knoten abzurufen.

Redis-Knoten und -Shards

Ein Shard (in der API und CLI: Knotengruppe) ist eine hierarchische Anordnung von Knoten, die jeweils in einem Cluster eingebunden 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. Redis Version 3.2 und höher unterstützen mehrere Shards innerhalb eines Clusters (in der API und CLI: eine Replikationsgruppe). Dieser Support ermöglicht die Partitionierung Ihrer Daten in einem Redis-Cluster (Cluster-Modus aktiviert).

Das folgende Diagramm veranschaulicht die Unterschiede zwischen einem Redis-Cluster (Cluster-Modus deaktiviert) und einem Redis-Cluster (Cluster-Modus aktiviert).



Redis-Cluster (Cluster-Modus aktiviert) unterstützen die Replikation über Shards. Die API-Operation [DescribeReplicationGroups](#) (CLI: [describe-replication-groups](#)) listet die Knotengruppen mit den Mitgliedsknoten, die Rolle des Knotens innerhalb der Knotengruppe und weitere Informationen auf.

Wenn Sie einen Redis-Cluster erstellen, können Sie festlegen, ob Sie einen Cluster mit aktiviertem Clustering erstellen möchten. Redis-Cluster (Cluster-Modus deaktiviert) haben nie mehr als einen Shard, der durch Hinzufügen (bis zu insgesamt fünf) oder Löschen von Lesereplikat-Knoten horizontal skaliert werden kann. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#), [Hinzufügen eines Lesereplikats für Redis-Replikationsgruppen \(Clustermodus deaktiviert\)](#) oder [Löschen eines Lesereplikats für Redis-Replikationsgruppen \(Clustermodus deaktiviert\)](#). Redis-Cluster (Cluster-Modus deaktiviert) können auch vertikal skaliert werden, indem

Knotentypen geändert werden. Weitere Informationen finden Sie unter [Skalieren von Redis-Clustern \(Clustermodus deaktiviert\) mit Replikatknoten](#).

Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Version der Redis-Engine 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. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht 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.

Nachdem ein Redis-Cluster (mit aktiviertem Cluster-Modus) erstellt wurde, kann er geändert (vergrößert oder verkleinert) werden. Weitere Informationen finden Sie unter [Skalierung ElastiCache für Redis](#) und [Ersetzen von Knoten](#).

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 finden Sie unter [Erstellen manueller Backups](#) und [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Mit Knoten verbinden

Bevor Sie versuchen, eine Verbindung mit den Knoten in Ihrem Redis-Cluster herzustellen, benötigen Sie die Endpunkte für die Knoten. Informationen zum Ermitteln der Endpunkte finden Sie unter:

- [Suche nach den Endpunkten eines Redis-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Suchen von Endpunkten für einen Redis-Cluster \(Cluster-Modus aktiviert\) \(Konsole\)](#)
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Finden von Endpunkten \(ElastiCache-API\)](#)

Im folgenden Beispiel verwenden Sie das `redis-cli`-Dienstprogramm, um eine Verbindung mit einem Cluster herzustellen, der Redis ausführt.

Note

Weitere Informationen über Redis und verfügbare Redis-Befehle finden Sie auf der Webseite <http://redis.io/commands>.

So stellen Sie eine Verbindung mit einem Redis-Cluster mithilfe des `redis-cli`-Dienstprogramms her

1. Verbinden Sie sich mit Ihrer Amazon-EC2-Instance mit dem Verbindungsdienstprogramm Ihrer Wahl.

Note

Anweisungen zur Verbindung mit einer Amazon-EC2-Instance finden Sie im [Amazon-EC2-Erste-Schritte-Handbuch](#).

2. Laden Sie zum Entwickeln von `redis-cli` die GNU Compiler Collection (`gcc`) herunter und installieren Sie diese. Geben Sie an der Befehlsaufforderung Ihrer EC2-Instance den folgenden Befehl und dann `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 redis-cli-Dienstprogramm herunter und kompilieren Sie es. Dieses Dienstprogramm ist Teil der Redis-Softwaredistribution. Geben Sie an der Eingabeaufforderung Ihrer EC2-Instance die folgenden Befehle ein:

 Note

Führen Sie bei Ubuntu-Systemen `make distclean` aus, bevor Sie `make` ausführen.

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

4. Geben Sie bei der Eingabeaufforderung Ihrer EC2-Instance den folgenden Befehl ein.

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

Es erscheint eine Redis-Eingabeaufforderung ähnlich der Folgenden.

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

5. Prüfen Sie die Verbindung mithilfe der Redis-Befehle.

Sie sind nun mit dem Cluster verbunden und können Redis-Befehle ausführen. Im Folgenden sehen Sie einige Beispielbefehle mit den zugehörigen Redis-Antworten.

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

                        // wait 5 seconds

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

Zum Herstellen der Verbindung zu Knoten oder Clustern mit SSL (Secure Sockets Layer)-Verschlüsselung (Verschlüsselung während der Übertragung aktiviert) siehe [ElastiCache Verschlüsselung bei der Übertragung \(TLS\)](#).

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 den Leistungsdetails der einzelnen Knotentypen finden Sie unter [Amazon-EC2-Instance-Typen](#).

Informationen darüber, welche Knotengröße zu verwenden ist, finden Sie unter [Auswahl der Knotengröße](#).

Aktuelle Generation

Weitere Informationen zur vorherigen Generation finden Sie unter [Knoten der vorherigen Generation](#).

Note

Instance-Typen mit bustfähiger Netzwerkleistung nutzen ein Netzwerk-E/A-Guthabenmechanismus, um ihre Basisbandbreite nach bestem Aufwand überschreiten können.

Allgemeines

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes E/A-Multiplexing (Redis 7.0.4+) | 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 |

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes Bandbreiten-E/A-Multiplexing (Redis 7.0.4+) | Baseline-Bandbreite (Gbit/s) | Maximale Bandbreite (Gbit/s) |
|------------------------|-----------------------------------|---------------------------------|-------------------------------|--|------------------------------|------------------------------|
| 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 |
| 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 |

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes Bandbreiten-E/A-Multiplexing (Redis 7.0.4+) | Baseline (Gbit/s) | Maximale Bandbreite (Gbit/s) |
|--------------------|-----------------------------------|---------------------------------|-------------------------------|--|-------------------|------------------------------|
| cache.m6g.12xlarge | 5.0.6 | Y | Y | Y | 20 | N/A |
| cache.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 |
| 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 |

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes Bandbreiten-E/A-Multiplexing (Redis 7.0.4+) | Baselines (Gbit/s) | Maximale Bandbreite (Gbit/s) |
|-------------------|-----------------------------------|---------------------------------|-------------------------------|--|--------------------|------------------------------|
| cache.m4.4xlarge | 3.2.4 | Y | Y | Y | 2.0 | 10.0 |
| cache.m4.10xlarge | 3.2.4 | Y | Y | Y | 5.0 | 10.0 |
| cache.t4g.micro | 3.2.4 | N | N | N | 0,064 | 5.0 |
| cache.t4g.small | 5.0.6 | N | N | N | 0,128 | 5.0 |
| cache.t4g.medium | 5.0.6 | N | N | N | 0,256 | 5.0 |
| cache.t3.micro | 3.2.4 | N | N | N | 0,064 | 5.0 |
| cache.t3.small | 3.2.4 | N | N | N | 0,128 | 5.0 |
| cache.t3.medium | 3.2.4 | N | N | N | 0,256 | 5.0 |
| cache.t2.micro | 3.2.4 | N | N | N | 0,064 | 1,024 |
| cache.t2.small | 3.2.4 | N | N | N | 0,128 | 1,024 |
| cache.t2.medium | 3.2.4 | N | N | N | 0,256 | 1,024 |

RAM-optimiert

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes Bandbreiten-E/A-Multi plexing (Redis 7.0.4+) | Baseline (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 |
| 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 |
| cache.r6g.2xlarge | 5.0.6 | Y | Y | Y | 2.5 | 10.0 |
| cache.r6g.4xlarge | 5.0.6 | Y | Y | Y | 5.0 | 10.0 |

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes Bandbreiten-E/A-Multiplexing (Redis 7.0.4+) | Baselinne (Gbit/s) | Maximale Bandbreite (Gbit/s) |
|--------------------|-----------------------------------|---------------------------------|-------------------------------|--|--------------------|------------------------------|
| cache.r6g.8xlarge | 5.0.6 | Y | Y | Y | 12 | N/A |
| cache.r6g.12xlarge | 5.0.6 | Y | Y | Y | 20 | N/A |
| 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 |

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes E/A-Multiplexing (Redis 7.0.4+) | Baselinete Bandbreite (Gbit/s) | Maximale Bandbreite (Gbit/s) |
|-------------------|-----------------------------------|---------------------------------|-------------------------------|--|--------------------------------|------------------------------|
| 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 Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes E/A-Multiplexing (Redis 7.0.4+) | Baselinete Bandbreite (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 |

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes E/A-Multiplexing (Redis 7.0.4+) | Baselines Bandbreite (Gbit/s) | Maximale Bandbreite (Gbit/s) |
|---------------------|-----------------------------------|---------------------------------|-------------------------------|--|-------------------------------|------------------------------|
| 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 Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes E/A-Multiplexing (Redis 7.0.4+) | Baselines Bandbreite (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 |

| Instance-Typ | Unterstützte Redis-Mindestversion | Erweiterte E/A (ab Redis 5.0.6) | TLS-Offloading (Redis 6.2.5+) | Verbessertes Bandbreiten-E/A-Multiplexing (Redis 7.0.4+) | Baseline-Bandbreite (Gbit/s) | Maximale Bandbreite (Gbit/s) |
|---------------------|-----------------------------------|---------------------------------|-------------------------------|--|------------------------------|------------------------------|
| 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 Details finden Sie unter [Amazon ElastiCache-Preise](#).

Instances mit Spitzenlastleistung

Sie können allgemeine burstfähige T4g-, T3-Standard- und T2-Standard-Cache-Knoten in Amazon ElastiCache starten. Diese Knoten bieten CPU-Baseline-Leistung und die Möglichkeit, diese Leistung jederzeit zu bursten, bis das angesammelte Guthaben aufgebraucht ist. Ein CPU-Guthaben stellt die Leistung eines gesamten CPU-Kerns für eine Minute zur Verfügung.

Die T4g-, T3- und T2-Knoten von Amazon ElastiCache sind als Standard konfiguriert und eignen sich für Workloads mit durchschnittlicher CPU-Auslastung, die konstant unter der Basisleistung der Instance liegt. Um die Baseline-Leistung zu überschreiten, verwendet der Knoten Guthaben, die als CPU-Guthaben angefallen sind. Wenn das beim Knoten angefallene Guthaben sinkt, wird die Leistung schrittweise bis auf Baseline-Leistung gesenkt. Dank dieser schrittweisen Absenkung kommt es beim Knoten nicht zu einem drastischen Leistungsabfall, wenn das CPU-Guthaben ausgeschöpft ist. Weitere Informationen finden Sie unter [CPU-Guthaben und Basisleistung für Burstable-Performance-Instances](#) im Benutzerhandbuch zu Amazon EC2.

Die folgende Tabelle listet die burst-fähigen Leistungsknotentypen und die Rate der verdienten CPU-Guthaben pro Stunde auf. Sie zeigt zudem die maximale Anzahl verdienter CPU-Guthaben, die bei einem Knoten anfallen können, sowie die Anzahl der vCPUs pro Knoten. Ergänzend wird die Baseline-Leistung als Prozentsatz der vollen Kernleistung (einer einzelnen vCPU) angegeben.

| Pro Stunde erworbenes CPU-Guthaben | Maximal verdiente Guthaben, die angesammelt werden können* | vCPUs | Basisleistung pro vCPU | 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 |
| 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 Basisleistung in der Tabelle ist pro vCPU. Bei einigen Knotengrößen kommen mehrere vCPUs zum Einsatz. Für diese wird die CPU-Baseline-Auslastung eines Knoten berechnet, indem Sie den vCPU-Prozentsatz mit der Anzahl der vCPUs multiplizieren.

Die folgenden CPU-Kreditmetriken sind für T3- und T4g-Instances mit Spitzenleistung verfügbar:

Note

Diese Metriken sind nicht für T2-Instances mit Spitzenleistung verfügbar.

- CPUCreditUsage
- CPUCreditBalance

Weitere Informationen zu diesen Metriken finden Sie unter [CPU-Guthaben-Metriken](#).

Darüber hinaus sollten Sie Folgendes beachten:

- Alle Knotentypen der aktuellen Generation werden standardmäßig in einer Virtual Private Cloud (VPC) auf der Grundlage von Amazon VPC erstellt.
- Redis-AOF (Append-Only-Files) werden für T2-Instances nicht unterstützt. Die Redis-Konfigurationsvariablen `appendonly` und `appendfsync` werden unter Redis ab Version 2.8.22 nicht unterstützt.

Verwandte Informationen

- [Amazon ElastiCache – Produktfunktionen und -details](#)
- [Redis-spezifische Parameter](#)
- [Verschlüsselung während der Übertragung](#)

Neustart von Knoten (nur mit deaktiviertem Cluster-Modus)

Einige Änderungen erfordern einen Neustart der Cluster-Knoten, damit die Änderungen übernommen werden können. Beispielsweise wird bei einigen Parametern eine Änderung am Parameterwert in einer Parametergruppe erst nach einem Neustart übernommen.

Dies sind die Parameter für Redis-Cluster (Cluster-Modus deaktiviert):

- activerehashing
- databases

Sie können einen Knoten nur mit der ElastiCache-Konsole neu starten. Sie können immer nur einen einzelnen Knoten neu starten. Um mehrere Knoten neu zu starten, müssen Sie den Vorgang für jeden Knoten wiederholen.

Redis-Parameteränderungen (Cluster Mode Aktiviert)

Führen Sie nachstehende Schritte aus, wenn Sie Änderungen an den folgenden Parametern in einem Redis-Cluster (Clustermodus aktiviert) vornehmen.

- activerehashing
 - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Redis-Cluster (Cluster-Modus aktiviert). Siehe [Löschen eines Clusters](#).
 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.

Verwendung von AWS Management Console

Sie können einen Knoten nur mit der ElastiCache-Konsole neu starten.

So starten Sie einen Knoten (Konsole) neu

1. Melden Sie sich bei AWS Management Console an 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 gewünschte AWS-Region aus.
3. Wählen Sie im linken Navigationsbereich Redis.

Eine Liste der Cluster, in denen Redis läuft, wird eingeblendet.

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.

Ersetzen von Knoten

Amazon ElastiCache for Redis aktualisiert seine Flotte regelmäßig, wobei Patches und Upgrades nahtlos auf die Instances angewendet werden. Von Zeit zu Zeit müssen wir jedoch Ihre ElastiCache for Redis-Knoten neu starten, um obligatorische Betriebssystem-Updates auf dem 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 ElastiCache automatisch generierte Ersatz-Cache-Knoten 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, wenn ElastiCache einen Ihrer Redis-Knoten zum Austausch vorsieht. Um die für Ihre Situation erforderlichen Informationen schneller zu finden, wählen Sie eine Option aus dem folgenden Menü aus.

- [Do nothing](#) – Lassen Sie Amazon ElastiCache den Knoten wie geplant ersetzen.
- [Change your maintenance window](#) – Ändern Sie Ihr Wartungsfenster und wählen Sie einen besser geeigneten Termin.
- Redis-Konfigurationen (Cluster-Modus aktiviert)
 - [Replace the only node in any Redis cluster](#) – Ein Verfahren zum Ersetzen eines Knotens in einem Redis-Cluster mithilfe von Sicherung und Wiederherstellung.
 - [Replace a replica node in any Redis cluster](#) – Ein Verfahren zum Ersetzen eines Lesereplikats in einem beliebigen Redis-Cluster durch Erhöhen und Verringern der Replikanzahl ohne Ausfallzeit des Clusters.

- [Replace any node in a Redis \(cluster mode enabled\) shard](#) – Ein dynamisches Verfahren ohne Ausfallzeit des Clusters zum Ersetzen eines Knotens in einem Redis-Cluster (Cluster-Modus aktiviert) durch Heraus- und Hineinskalieren.
- Redis-Konfigurationen (Cluster-Modus deaktiviert)
 - [Replace the only node in any Redis cluster](#) – Verfahren zum Ersetzen eines beliebigen Knotens in einem Redis-Cluster mithilfe von Sicherung und Wiederherstellung.
 - [Replace a replica node in any Redis cluster](#) – Ein Verfahren zum Ersetzen eines Lesereplikats in einem beliebigen Redis-Cluster durch Erhöhen und Verringern der Replikanzahl ohne Ausfallzeit des Clusters.
 - [Replace a node in a Redis \(cluster mode disabled\) cluster](#) – Verfahren zum Ersetzen eines Knotens in einem Redis-Cluster (Cluster-Modus deaktiviert) mithilfe der Replikation.
 - [Replace a Redis \(cluster mode disabled\) read-replica](#) – Ein Verfahren zum manuellen Ersetzen eines Lesereplikats in einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert).
 - [Replace a Redis \(cluster mode disabled\) primary node](#) – Ein Verfahren zum manuellen Ersetzen eines Primärknotens in einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert).

Optionen für das Ersetzen von Redis-Knoten

- Keine Aktion – Wenn Sie nichts tun, ersetzt ElastiCache den Knoten wie geplant.

Bei Nicht-Cluster-Konfigurationen mit aktiviertem Autofailover schließen Cluster auf Redis 5.0.6 und höher den Austausch ab, während der Cluster weiterhin online bleibt und eingehende Schreibanforderungen bedient. Bei Clustern auf Redis 4.0.10 oder niedriger, für die automatisches Failover aktiviert ist, stellen Sie im Zusammenhang mit dem DNS-Update möglicherweise eine kurze Schreibunterbrechung von bis zu einigen Sekunden fest.

Wenn der Knoten ein Mitglied eines Clusters mit automatischem Failover ist, bietet ElastiCache for Redis eine verbesserte Verfügbarkeit während Patches, Updates und anderen wartungsbedingten Knotenaustauschvorgängen.

Für ElastiCache für Redis-Cluster-Konfigurationen, die für die Verwendung von ElastiCache für Redis-Cluster-Clients eingerichtet wurden, wird der Austausch nun durchgeführt, während der Server eingehende Schreibanforderungen ausführt.

Bei Nicht-Cluster-Konfigurationen mit aktiviertem Autofailover schließen Cluster auf Redis 5.0.6 und höher den Austausch ab, während der Cluster weiterhin online bleibt und eingehende Schreibanforderungen bedient. Bei Clustern auf Redis 4.0.10 oder niedriger, für die automatisches Failover aktiviert ist, stellen Sie im Zusammenhang mit dem DNS-Update möglicherweise eine kurze Schreibunterbrechung von bis zu einigen Sekunden fest.

Wenn es sich um einen eigenständigen Knoten handelt, startet Amazon ElastiCache zunächst einen Ersatzknoten und führt dann eine Synchronisierung mit dem vorhandenen Knoten durch. Der vorhandene Knoten ist während dieser Zeit nicht für Serviceanfragen verfügbar. Sobald die Synchronisierung abgeschlossen ist, wird der vorhandene Knoten beendet und der neue Knoten nimmt seinen Platz ein. ElastiCache bemüht sich nach Kräften, Ihre Daten während dieses Vorgangs zu erhalten.

- Ändern Sie Ihr Wartungsfenster – Bei geplanten Wartungsereignissen erhalten Sie eine E-Mail oder ein Benachrichtigungsereignis 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, das Austauschfenster durch Verschieben des Wartungsfensters zu ändern, ist nur verfügbar, wenn die ElastiCache-Benachrichtigung ein Wartungsfenster enthält. 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 Wartung](#).

- Ersetzen des einzigen Knotens in einem Redis-Cluster – Wenn der Cluster keine Lesereplikate hat, können Sie das folgende Verfahren verwenden, um den Knoten zu 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](#).
 4. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.
- Ersetzen eines Replikationsknotens in einem beliebigen Redis-Cluster – Um einen Replikationscluster zu ersetzen, erhöhen Sie die Anzahl der Replikate. Fügen Sie dafür ein Replikat hinzu und verringern Sie die Replikanzahl, indem Sie das Replikat 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.

So ersetzen Sie ein Replikat in einem beliebigen Redis-Cluster

1. Erhöhen Sie die Replikanzahl, indem Sie ein Replikat zum Shard oder zur Replikationsgruppe hinzufügen. Weitere Informationen finden Sie unter [Erhöhen der Anzahl der Replikate in einem Shard](#).
 2. Löschen Sie das Replikat, 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 eines beliebigen Knotens in einem Redis-Shard (Cluster-Modus aktiviert) – Um den Knoten in einem Cluster ohne Ausfallzeit zu ersetzen, verwenden Sie Online-Resharding. Fügen Sie zunächst einen Shard durch Ausskalieren hinzu und löschen Sie den Shard mit dem Knoten, der durch Einkalieren ersetzt werden soll.

So ersetzen Sie einen beliebigen Knoten in einem Redis-Cluster (Cluster-Modus aktiviert)

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 eines Knotens in einem Redis-Cluster (Cluster-Modus deaktiviert) – Wenn der Cluster ein Redis-Cluster (Cluster-Modus deaktiviert) ohne Lesereplikate ist, 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 [So fügen Sie zu einem Redis-Cluster ohne Shards die Replikation hinzu](#).
 2. Fügen Sie dem Cluster ein Lesereplikat hinzu. Detaillierte Anweisungen finden Sie unter [So fügen Sie Knoten zu einem Cluster hinzu \(Konsole\)](#).
 3. Stufen Sie das soeben erstellte Lesereplikat auf Primär hoch. Detaillierte Anweisungen finden Sie unter [Redis \(Cluster-Modus deaktiviert\) -Replikationsgruppen](#).
 4. Löschen Sie den für das Ersetzen geplanten Knoten. Detaillierte Anweisungen finden Sie unter [Entfernen von Knoten aus einem Cluster](#).
 5. Ersetzen Sie in Ihrer Anwendung den Endpunkt des alten durch den Endpunkt des neuen Knotens.
- Ersetzen eines Redis-Lesereplikats (Cluster-Modus deaktiviert) – Wenn der Knoten ein Lesereplikat ist, 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](#).

So ersetzen Sie ein Redis-Lesereplikat (Cluster-Modus deaktiviert)

1. Löschen Sie das für den Austausch geplante Replikat. Detaillierte Informationen finden Sie hier:
 - [Verringern der Anzahl der Replikate in einem Shard](#)
 - [Entfernen von Knoten aus einem Cluster](#)
2. Fügen Sie ein neues Replikat hinzu, das das auszutauschende Replikat ersetzen soll. Wenn Sie denselben Namen wie für das soeben gelöschte Replikat verwenden, können Sie Schritt 3 überspringen. Detaillierte Informationen finden Sie hier:
 - [Erhöhen der Anzahl der Replikate in einem Shard](#)
 - [Hinzufügen eines Lesereplikats für Redis-Replikationsgruppen \(Clustermodus deaktiviert\)](#)
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](#).
- Ersetzen eines Redis-Primärknotens (Cluster-Modus deaktiviert) – Wenn der Knoten der Primärknoten ist, befördern Sie zunächst ein Lesereplikat zum Primärknoten. Löschen Sie dann das Replikat, das vorher der primäre Knoten war.

Wenn Ihr Cluster nur ein Replikat enthält und Multi-AZ aktiviert ist, müssen Sie Multi-AZ deaktivieren, bevor Sie das Replikat in Schritt 2 löschen können. Detaillierte Anweisungen finden Sie unter [Ändern einer Replikationsgruppe](#).

So ersetzen Sie einen Redis-Primärknoten (Cluster-Modus deaktiviert)

1. Stufen Sie ein Lesereplikat auf Primär hoch. Detaillierte Anweisungen finden Sie unter [Redis \(Cluster-Modus deaktiviert\) -Replikationsgruppen](#).
2. Löschen Sie den Knoten, der ausgetauscht werden soll (der alte primäre Knoten). Detaillierte Anweisungen finden Sie unter [Entfernen von Knoten aus einem Cluster](#).
3. Fügen Sie ein neues Replikat hinzu, das das auszutauschende Replikat 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 eines Lesereplikats für Redis-Replikationsgruppen \(Clustermodus deaktiviert\)](#).

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](#).

Reservierte Knoten für ElastiCache

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.

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. Information zu den Preisen finden auf der Seite [Preise für Amazon ElastiCache](#).

Note

Reservierte Knoten sind nicht flexibel, sie gelten nur für den genauen Instance-Typ, den Sie reservieren.

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. Information zu den Preisen finden auf der Seite [Preise für Amazon ElastiCache](#).

Die AWS-Region, der Knotentyp und die Laufzeit müssen beim Kauf ausgewählt werden und können später nicht mehr geändert werden.

Sie können die AWS Management Console, die AWS CLI oder die ElastiCache-API verwenden, um verfügbare Angebote für reservierte Knoten aufzulisten und diese Knoten zu erwerben.

Weitere Informationen zu reservierten Knoten finden Sie unter [Reservierte Knoten von Amazon ElastiCache](#).

Themen

- [Standardangebote für reservierte Knoten](#)
- [Ältere Angebote für reservierte Knoten](#)
- [Abrufen von Informationen über Angebote für reservierte Knoten](#)
- [Kauf eines reservierten Knotens](#)
- [Abrufen von Informationen über Ihre reservierten Knoten](#)

Standardangebote für reservierte Knoten

Wenn Sie eine standardmäßige reservierte Knoten-Instance (RI) in Amazon ElastiCache kaufen, erwerben Sie damit die Zusage, für die Dauer der reservierten Knoten-Instance einen ermäßigten Tarif für einen bestimmten Cache-Knoten-Instance-Typ und eine bestimmte AWS-Region zu erhalten. Zum Verwenden einer reservierten Knoten-Instance für Amazon ElastiCache erstellen Sie eine neue ElastiCache-Knoten-Instance, genau wie bei einer On-Demand-Instance.

Die neue Knoten-Instance muss exakt mit den Spezifikationen der reservierten Knoten-Instance übereinstimmen. Wenn die Spezifikationen der neuen Knoten-Instance mit denen einer vorhandenen reservierten Knoten-Instance Ihres Kontos übereinstimmen, wird Ihnen der reduzierte Preis für die reservierte Knoten-Instance in Rechnung gestellt. Andernfalls wird die Knoten-Instance auf Basis eines On-Demand-Preises abgerechnet. Diese standardmäßigen Reserved Instances sind ab den Instance-Familien R5 und M5 verfügbar.

Note

Alle im Folgenden besprochenen Angebotstypen stehen mit einer Laufzeit von einem Jahr und einer Laufzeit von drei Jahren zur Verfügung.

Angebotstypen

Keine Vorauszahlung RI bietet Zugriff auf eine ElastiCache-Instance, ohne dass eine Vorauszahlung erforderlich ist. Für Ihre ohne Vorauszahlung reservierte ElastiCache-Instance wird für jede Stunde innerhalb der Laufzeit ein ermäßigter Stundensatz abgerechnet, unabhängig von der Nutzung. Es ist keine Vorauszahlung erforderlich.

Teilweise Vorauszahlung: Bei dieser Option wird ein Teil der reservierten ElasticCache-Instance im Voraus bezahlt. 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.

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 | Medium | 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 Nutzungsgebühr pro Stunde, wenn | Die höchsten Gesamtpreise, wenn Sie planen, die ganze Zeit in Betrieb zu sein. Dies sind jedoch die niedrigsten Gesamtpreise, |

| Angebot | Vorabkosten | Nutzungsgebühr | Vorteil |
|--|-------------|---|--|
| | | der Knoten nicht ausgeführt wird. Höchste Gebühren pro Stunde aller Angebotstarife, die Gebühren fallen jedoch nur an, wenn der reservierte Knoten ausgeführt wird. | 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) | Keine | Höchste Gebühr pro Stunde. Fällt nur immer dann an, wenn der Knoten ausgeführt wird. | Höchste Kosten pro Stunde. |

Weitere Informationen finden Sie unter [Amazon ElastiCache-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 die Preise und Informationen über verfügbare reservierte Knotenangebote mithilfe der AWS Management Console, AWS CLI und ElastiCache-API abgerufen werden können.

Themen

- [Abrufen von Informationen über Angebote für reservierte Knoten \(Konsole\)](#)
- [Abrufen von Informationen über Angebote für reservierte Knoten \(AWS CLI\)](#)
- [Abrufen von Informationen zu reservierten Knotenangeboten \(ElastiCache API\)](#)

Abrufen von Informationen über Angebote für reservierte Knoten (Konsole)

Führen Sie die folgenden Schritte aus, um mithilfe der AWS Management Console Preisinformationen sowie weitere Informationen zu verfügbaren Angeboten für reservierte Cluster zu erhalten.

So erhalten Sie Informationen zu verfügbaren Angeboten für reservierte Knoten

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Reserved Nodes (Reservierte Knoten) aus.
3. Wählen Sie Purchase Reserved Node (Reservierten Knoten kaufen) aus.
4. Wählen Sie für Engine Redis 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 Gesamtkosten der ausgewählten Optionen unter Reservation details (Reservierungsdetails) angezeigt.

6. Wählen Sie Cancel, um den Kaufvorgang für diese Knoten abzubrechen und Kosten zu vermeiden.

Abrufen von Informationen über Angebote für reservierte Knoten (AWS CLI)

Zum Abrufen von Preisinformationen sowie weiteren Informationen zu verfügbaren Angeboten für reservierte Knoten, geben Sie den folgenden Befehl an einer Befehlsaufforderung ein:

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

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

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

Weitere Informationen finden Sie unter [describe-reserved-cache-nodes-offerings](#) in der AWS CLI-Referenz.

Abrufen von Informationen zu reservierten Knotenangeboten (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 unter [DescribeReservedCacheNodesOfferings](#) in der ElastiCache-API-Referenz.

Kauf eines reservierten Knotens

Die folgenden Beispiele zeigen, wie ein reserviertes Knotenangebot mit der AWS Management Console, der AWS CLI und der ElastiCache-API erworben werden kann.

Important

Wenn Sie die Beispiele in diesem Abschnitt befolgen, wird Ihr AWS-Konto mit Gebühren belastet, 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.

Im folgenden Verfahren wird die AWS Management Console verwendet, um einen reservierten Knoten mit der Angebots-ID zu kaufen.

So kaufen Sie reservierte Knoten

1. Melden Sie sich bei 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 Redis 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 unter [PurchaseReservedCacheNodesOffering](#) in der ElastiCache-API-Referenz.

Abrufen von Informationen über Ihre reservierten Knoten

Sie können Informationen über die reservierten Knoten, die Sie erworben haben, mit der AWS Management Console, der AWS CLI und der ElastiCache-API abrufen.

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 der AWS Management Console Informationen über die reservierten Knoten abrufen, die Sie gekauft haben.

So rufen Sie Informationen über Ihre gekauften reservierten Knoten ab

1. Melden Sie sich bei 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.

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 den folgenden Befehl an einer Eingabeaufforderung ein:

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

Dieser Vorgang erzeugt eine Ausgabe ähnlich der folgenden (JSON-Format):

```
{
  "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)

Um Informationen über reservierte Knoten für Ihr AWS-Konto zu erhalten, rufen Sie den DescribeReservedCacheNodes-Vorgang auf.

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 unter [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 bestehenden Cluster haben, die einen Knotentyp der vorherigen Generation verwenden, unterstützt ElastiCache nicht die Erstellung neuer Cluster mit diesem Knotentyp.

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 in ElastiCache finden Sie unter [Unterstützte Knotentypen](#).

Migrieren von Knoten auf einem Redis-Cluster

Die folgende Vorgehensweise beschreibt, wie Sie Ihren Redis-Cluster-Knotentyp mit Hilfe der ElastiCache-Konsole migrieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten. Abhängig von Ihrer Clusterkonfiguration können folgende Ausfallzeiten auftreten. Die folgenden Angaben sind Schätzungen und können je nach Ihrer spezifischen Konfiguration abweichen:

- Bei deaktiviertem Cluster-Modus (Einzelknoten) kann es zu einer Verzögerung von ca. 60 Sekunden kommen, die in erster Linie auf die DNS-Verbreitung zurückzuführen ist.
- Bei deaktiviertem Cluster-Modus (mit Replikationsknoten) kann bei Clustern mit Redis 5.0.6 und höher etwa 1 Sekunde angezeigt werden. Alle niedrigeren Versionen können etwa 10 Sekunden erwarten.
- Bei aktiviertem Cluster-Modus kann etwa 1 Sekunde angezeigt werden.

Um einen Redis-Cluster-Knotentyp über die Konsole zu ändern:

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 Redis clusters (Redis-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 Redis-Cluster-Knotentyp zu ändern, verwenden Sie die AWS CLI:

Verwenden Sie die API [modify-replication-group](#) wie im Folgenden gezeigt:

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, auf den migriert wird. 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 Knoten mit dem Status „restore-failed“

Das folgende Verfahren beschreibt, wie Sie einen oder mehrere Knoten mit dem Status „restore-failed“ reparieren oder aus Ihrem Redis-Cluster entfernen. Weitere Informationen darüber, wie ElastiCache-Knoten in den Status „restore-failed“ übergehen, finden Sie unter [ElastiCache Knotenstatus anzeigen](#). Wir empfehlen, zuerst alle Knoten mit dem Status „restore-failed“ zu entfernen, 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 ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis clusters (Redis-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 den bzw. die Knoten mit dem Status „restore-failed“ aus Ihrem ElastiCache-Cluster entfernt haben, können Sie jetzt zu einem Knotentyp der neueren Generation migrieren. Weitere Informationen finden Sie weiter oben unter [Migrieren von Knoten auf einem Redis-Cluster](#).

Informationen zum Hinzufügen von Back-Nodes zu Ihrem ElastiCache-Cluster finden Sie unter [Knoten zu einem Cluster hinzufügen](#).

Verwalten von Clustern

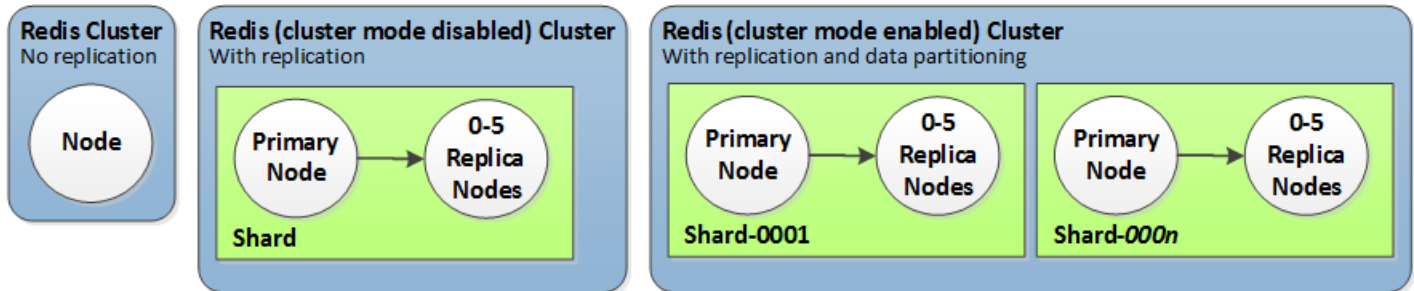
Ein Cluster ist eine Sammlung einer oder mehrerer Cache-Knoten, die alle eine Instance der unterstützten Cache-Engine-Software oder Redis ausführen. Wenn Sie einen Cluster erstellen, geben Sie die Engine und die Version an, die von allen Knoten verwendet werden soll.

Das folgende Diagramm zeigt einen typischen Redis-Cluster. Redis-Cluster können einen einzelnen Knoten oder bis zu sechs Knoten innerhalb eines Shards (API/CLI: Knotengruppe) enthalten. Ein Redis-Cluster mit einem Knoten (Cluster-Modus deaktiviert) hat keinen Shard, und ein Redis-Cluster mit mehreren Knoten (Cluster-Modus deaktiviert) hat einen einzelnen Shard. Redis-Cluster (Cluster-Modus aktiviert) können bis zu 500 Shards haben, wobei Ihre Daten auf die Shards verteilt sind. Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Version der Redis-Engine 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. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht 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 mehrere Knoten in einem Shard besitzen, ist einer der Knoten der primäre Knoten für Lese- und Schreibvorgänge. Alle anderen Knoten im Shard sind schreibgeschützte Replikate.

Typische Redis-Cluster sehen wie folgt aus.



Die meisten ElastiCache Operationen werden auf Cluster-Ebene 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 einen bestimmten Cluster an, wenn mit den ElastiCache -API- AWS CLI Befehlen und interagiert wird. 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 Amazon EC2 konzipiert. Wenn Sie Ihren Cluster in einer Virtual Private Cloud (VPC) starten, die auf dem Amazon VPC-Service basiert, können Sie von außerhalb AWS auf ihn zugreifen. Weitere Informationen finden Sie unter [Zugriff auf ElastiCache-Ressourcen außerhalb von AWS](#).

Eine Liste der unterstützten Redis-Versionen finden Sie unter [Unterstützte ElastiCache-for-Redis-Versionen](#).

Auswählen eines Netzwerktyps

ElastiCache unterstützt die Internetprotokollversionen 4 und 6 (IPv4 und IPv6). Dadurch können Sie Ihren Cluster so konfigurieren, dass er Folgendes akzeptiert:

- nur IPv4-Verbindungen,
- nur IPv4-Verbindungen,
- IPv4- und IPv6-Verbindungen (Dual-Stack-Verbindungen)

IPv6 wird für Workloads unterstützt, die die Redis-Engine ab Version 6.2 auf allen Instances verwenden, die auf dem [Nitro-System](#) erstellt wurden. Für den Zugriff auf ElastiCache über IPv6 fallen keine zusätzlichen Gebühren an.

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.

Konfigurieren von Subnetzen für den Netzwerktyp

Wenn Sie einen -Cluster in einer Amazon VPC erstellen, müssen Sie eine Subnetzgruppe angeben. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen und diese mit Ihrem Knoten zu verknüpfen. ElastiCache-Cluster benötigen ein Dual-Stack-Subnetz, dem sowohl IPv4- als auch IPv6-Adressen zugewiesen sind, um im Dual-Stack-Modus zu arbeiten, und ein reines IPv6-Subnetz, um nur als IPv6 zu arbeiten.

Verwenden von Dual-Stack

Wenn ElastiCache für Redis im aktivierten Cluster-Modus verwendet wird, unterscheidet sich aus Sicht einer Anwendung die Verbindung mit allen Cluster-Knoten über den Konfigurationsendpunkt nicht von der direkten Verbindung mit 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.

Zur Gewährleistung der Abwärtskompatibilität mit allen vorhandenen Clients wurde IP-Erkennung eingeführt. Dadurch können Sie den IP-Typ (d. h. IPv4 oder IPv6) auswählen, der im Erkennungsprotokoll angekündigt werden soll. Dies beschränkt die automatische Erkennung zwar auf nur einen IP-Typ, aber Dual-Stack ist für Workloads mit aktivierten Cluster-Modus dennoch von Vorteil, da es Migrationen (oder Rollbacks) von einem IPv4- zu einem IPv6-Discovery-IP-Typ ohne Ausfallzeiten ermöglicht.

TLS-fähige Dual-Stack-ElastiCache-Cluster

Wenn TLS für ElastiCache-Cluster aktiviert ist, geben die Clustererkennungsfunktionen (`cluster slots`, `cluster shards` und `cluster nodes`) Hostnamen statt IPs zurück. Die Hostnamen werden dann anstelle von IPs verwendet, um eine Verbindung mit dem ElastiCache-Cluster herzustellen und einen TLS-Handshake durchzuführen. Das bedeutet, dass Clients nicht vom IP-Discovery-Parameter betroffen sind. Bei TLS-fähigen Clustern hat der IP Discovery-Parameter keine Auswirkung auf das bevorzugte IP-Protokoll. Stattdessen wird das verwendete IP-Protokoll dadurch 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 [TLS-fähige Dual-Stack-Cluster ElastiCache](#).

Verwendung der AWS Management Console

Wählen Sie beim Erstellen eines Clusters mit der AWS Management Console unter Connectivity (Konnektivität) einen Netzwerktyp aus, entweder IPv4, IPv6 oder Dual stack (Dual-Stack). Wenn Sie einen Redis-Cluster (Cluster-Modus aktiviert) erstellen und Dual-Stack auswählen, müssen Sie anschließend einen Discovery IP type (Erkennungs-IP-Typ) auswählen, entweder IPv6 oder IPv4.

Weitere Informationen finden Sie unter [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#) oder [Einen Redis erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Wählen Sie beim Erstellen einer Replikationsgruppe mit dem AWS Management Console einen Netzwerktyp aus, entweder IPv4, IPv6 oder Dual-Stack. Wenn Sie sich für dual stack (Dual-Stack) entscheiden, müssen Sie einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Weitere Informationen finden Sie unter [Erstellen einer Redis-Replikationsgruppe \(Clustermodus deaktiviert\) von Grund auf](#) oder [Erstellen einer Replikationsgruppe in Redis \(Clustermodus aktiviert\) von Grund auf](#).

Verwenden der -CLI

Wenn Sie einen Cache-Cluster mit der CLI erstellen, verwenden Sie den Befehl [create-cache-cluster](#) und geben Sie die Parameter IPDiscovery sowie NetworkType an:

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 eine Replikationsgruppe mit deaktiviertem Cluster-Modus mithilfe der CLI erstellen, verwenden Sie den Befehl [create-replication-group](#) 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 Cluster-Modus mithilfe der CLI erstellen und IPv4 für die IP-Erkennung nutzen, verwenden Sie den Befehl [create-replication-group](#) und geben die Parameter `NetworkType` und `IPDiscovery` 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 Cluster-Modus mithilfe der CLI erstellen und IPv6 für die IP-Erkennung nutzen, verwenden Sie den Befehl [create-replication-group](#) und geben die Parameter `NetworkType` und `IPDiscovery` 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
```

Daten-Tiering

Die Daten von Clustern, die eine Replikationsgruppe umfassen und einen Knotentyp aus der R6gd-Familie verwenden, sind zwischen Speicher und lokalem SSD-Speicher (Solid-State-Laufwerke) gestaffelt. Daten-Tiering bietet eine neue Option mit gutem Preis-Leistungs-Verhältnis für Redis-Workloads, indem Daten nicht nur im Speicher gespeichert werden, sondern zusätzlich auch kostengünstigere Solid-State-Laufwerke (SSDs) in jedem Clusterknoten verwendet werden. Diese Option ist ideal für Workloads, die regelmäßig auf bis zu 20 Prozent ihres gesamten Datensatzes zugreifen, und für Anwendungen, die beim Zugriff auf Daten auf SSD zusätzliche Latenz tolerieren können.

ElastiCache überwacht auf Clustern mit Daten-Tiering die letzte Zugriffszeit jedes gespeicherten Elements. Wenn der verfügbare Speicher (DRAM) vollständig verbraucht ist, verwendet es einen LRU-Algorithmus (Least-Recently Used), um Objekte, auf die selten zugegriffen wird, automatisch vom Arbeitsspeicher auf die SSD zu verschieben. Wenn anschließend auf Daten auf der SSD zugegriffen 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 Daten-Tiering die Schlüssel selbst immer im Speicher verbleiben, während die LRU die Platzierung von Werten in dem Speicher im Vergleich zur Festplatte regelt. 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. Bei Annahme von 500-Byte-String-Werten können Sie beispielsweise durchschnittlich zusätzliche 300 Mikrosekunden Latenzzeit für Anforderungen an auf SSD gespeicherte Daten im Vergleich zu Anforderungen an Daten im Speicher erwarten.

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 kompatibel mit allen Redis-Befehlen und Datenstrukturen, 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:

- Daten-Tiering ist ideal für Workloads, die regelmäßig auf bis zu 20 Prozent ihres gesamten Datensatzes zugreifen, und für Anwendungen, die beim Zugriff auf Daten auf SSD zusätzliche Latenz tolerieren können.
- Bei Verwendung von SSD-Kapazität, die auf Daten-Tiering-Knoten verfügbar ist, empfehlen wir, dass die Wertgröße größer als die Schlüsselgröße ist. Wenn Elemente zwischen DRAM und SSD verschoben werden, 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, eu-north-1, eu-west-3, ap-northeast-1, ap-southeast-1, ap-southeast-2, ap-south-1, ca-central-1 und sa-east-1.
- Sie müssen die Redis-Engine Version 6.2 oder höher verwenden.
- Sie können ein Backup eines R6gd-Clusters nur dann in einem anderen Cluster wiederherstellen, wenn dieses ebenfalls R6gd verwendet.
- Backups für Cluster mit Daten-Tiering können nicht nach Amazon S3 exportiert werden.
- Die Online-Migration wird für Cluster mit R6gd-Knotentyp nicht unterstützt.
- Die Skalierung von einem Cluster mit Daten-Tiering (z. B. ein Cluster, der einen R6gd-Knotentyp verwendet) zu einem Cluster ohne Daten-Tiering (z. B. ein Cluster, der einen R6g-Knotentyp verwendet) wird nicht unterstützt. Weitere Informationen finden Sie unter [Skalierung ElastiCache für Redis](#).
- Auto Scaling wird auf Clustern unterstützt, die Daten-Tiering für Redis Version 7.0.7 und höher verwenden. Weitere Informationen finden Sie unter [Auto Scaling ElastiCache für Redis-Cluster](#).
- Daten-Tiering unterstützt nur die maxmemory-Richtlinien volatile-lru, allkeys-lru, volatile-lfu, allkeys-lfu und noeviction.
- Forkless Save wird für Redis Version 7.0.7 und höher unterstützt. Weitere Informationen finden Sie unter [So werden Synchronisation und Backup implementiert](#).
- Elemente, die größer als 128 MiB sind, werden nicht auf SSD verschoben.

Preisgestaltung

R6gd-Knoten haben eine 4,8-mal höhere Gesamtkapazität (Speicher + SSD). Bei maximaler Auslastung können Sie durch ihren Einsatz Einsparungen von über 60 Prozent im Vergleich zu R6g-Knoten (nur Speicher) erzielen. [Weitere Informationen finden Sie unter ElastiCache Preise.](#)

Überwachen

ElastiCache for Redis bietet Metriken, die speziell für die Überwachung der Leistungscluster entwickelt wurden, die Data Tiering verwenden. Um das Verhältnis von Elementen in DRAM im Vergleich zu SSD zu überwachen, können Sie die Metrik `currItems` unter [Metrics for Redis](#) (Metriken für Redis) verwenden. Sie können den Prozentsatz wie folgt berechnen: (CurrItems mit Dimension: Tier = Memory * 100)/(CurrItems ohne Dimensionsfilter).

Wenn die konfigurierte Löschrictlinie dies zulässt ElastiCache , beginnt Redis mit dem Löschen von Elementen, 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 [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#). Weitere Informationen zu Metriken für Redis-Cluster, die Data Tiering verwenden, finden Sie unter [Metriken für Redis](#)

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 [Erstellen eines Clusters](#).

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 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
```

Für Windows:

```
aws elasticache create-replication-group ^  
  --replication-group-id redis-dt-cluster ^  
  --replication-group-description "Redis cluster with data tiering" ^  
  --num-node-groups 1 ^  
  --replicas-per-node-group 1 ^  
  --cache-node-type cache.r6gd.xlarge ^  
  --engine redis ^  
  --cache-subnet-group-name default ^  
  --automatic-failover-enabled ^  
  --data-tiering-enabled
```

Nach dem Ausführen dieses Vorgangs sehen Sie eine Antwort ähnlich dem folgenden:

```
{  
  "ReplicationGroup": {  
    "ReplicationGroupId": "redis-dt-cluster",  
    "Description": "Redis cluster with data tiering",  
    "Status": "creating",  
    "PendingModifiedValues": {},  
    "MemberClusters": [  
      "redis-dt-cluster"  
    ],  
    "AutomaticFailover": "enabled",
```

```
    "DataTiering": "enabled",
    "SnapshotRetentionLimit": 0,
    "SnapshotWindow": "06:00-07:00",
    "ClusterEnabled": false,
    "CacheNodeType": "cache.r6gd.xlarge",
    "TransitEncryptionEnabled": false,
    "AtRestEncryptionEnabled": false
  }
}
```

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 der (Konsole), () oder (API) wiederherstellen. AWS CLI ElastiCache 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. (Cluster-Modus aktiviert (aufskalieren) – Wählen Sie dies für einen Redis-Cluster (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 sich im neuen Cluster (API/ CLI: Knotengruppen) befinden sollen.
 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. VPC auswählen – Wählen Sie die VPC aus, in der dieser Cluster erstellt werden soll.
 - 10 Parametergruppe – Wählen Sie die Parametergruppe aus, die ausreichend Speicher für den Redis-Overhead des ausgewählten Knotentyps reserviert.
6. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Erstellen.

Weitere Informationen zum Erstellen von Clustern finden Sie unter [Erstellen eines Clusters](#).

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 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 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 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
  }
}
```

Vorbereitung eines Clusters

Nachfolgend finden Sie Anleitungen zur Erstellung eines Clusters über die ElastiCache-Konsole, die AWS CLI oder die ElastiCache-API.

Sie können auch einen ElastiCache-Cluster erstellen, indem Sie [AWS CloudFormation](#) nutzen.

Weitere Informationen finden Sie unter [AWS::ElastiCache::CacheCluster](#) im Benutzerhandbuch zu AWS Cloud Formation, das eine Anleitung für die 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

- [Bestimmung Ihrer Anforderungen](#)
- [Auswahl der Knotengröße](#)

Bestimmung Ihrer Anforderungen

Vorbereitung

Wenn Sie die Antworten auf die folgenden Fragen kennen, können Sie die Erstellung Ihres Clusters vereinfachen:

- 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 Virtual Private Cloud (VPC) auf der Basis von Amazon VPC starten?

Important

Wenn Sie den Cluster in einer VPC starten, müssen Sie in derselben VPC eine Subnetzgruppe erstellen, bevor mit dem Erstellen eines Clusters beginnen. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#).

ElastiCache ist für den Zugriff von innerhalb AWS mit Amazon EC2 konzipiert. Wenn Sie jedoch in einer VPC starten, die auf Amazon VPC basiert, und Ihr Cluster sich in einer VPC befindet, können Sie den Zugriff von außerhalb AWS ermöglichen. Weitere Informationen finden Sie unter [Zugriff auf ElastiCache-Ressourcen außerhalb von AWS](#).

- Müssen Sie irgendwelche Parameterwerte anpassen?

Erstellen Sie in diesem Fall eine benutzerdefinierte Parametergruppe. Weitere Informationen finden Sie unter [Erstellen einer Parametergruppe](#).

Wenn Sie Redis verwenden, sollten Sie die Einstellung `reserved-memory` oder `reserved-memory-percent` verwenden. Weitere Informationen finden Sie unter [Verwalten von reserviertem Speicher](#).

- Müssen Sie eine eigene VPC-Sicherheitsgruppe erstellen?

Weitere Informationen finden Sie unter [Sicherheit in Ihrer VPC](#).

- Haben Sie vor, Fehlertoleranz zu implementieren?

Weitere Informationen finden Sie unter [Minimieren von Ausfällen](#).

Themen

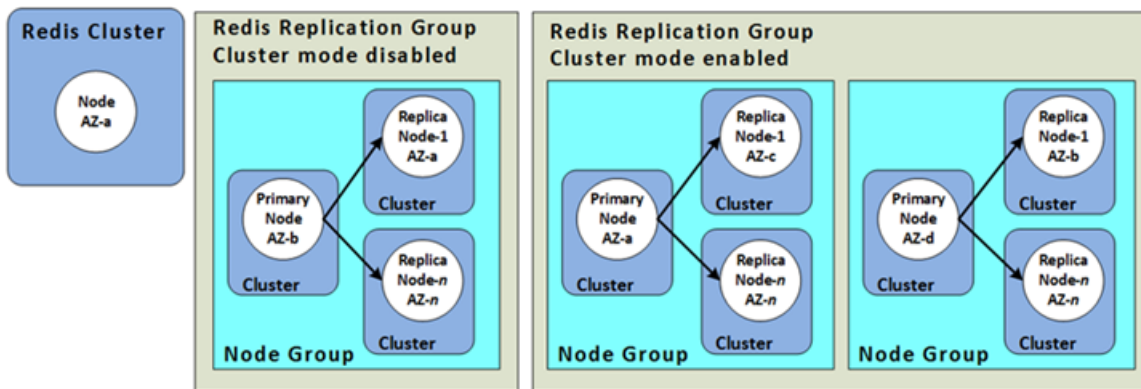
- [Anforderungen an Speicher und Prozessor](#)
- [Konfiguration von Redis-Cluster](#)
- [Voraussetzungen für Skalierung](#)
- [Zugriffsvoraussetzungen](#)
- [Anforderungen für Region und Availability Zone](#)

Anforderungen an Speicher und Prozessor

Der Grundbaustein für Amazon ElastiCache ist der Knoten. 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.

Konfiguration von Redis-Cluster

ElastiCache für Redis-Cluster bestehen aus 0 bis 500 Shards (auch Knotengruppen genannt). Die Daten in einem Redis-Cluster werden über die Shards in dem Cluster hinweg partitioniert. Ihre Anwendung stellt über eine als Endpunkt bezeichnete Netzwerkadresse eine Verbindung mit einem Redis-Cluster her. Die Knoten in einem Redis-Shard erfüllen eine von zwei Rollen: ein Lese/Schreib-Primärknoten und alle anderen Lese-Sekundärknoten (auch Read-Replicas genannt). Zusätzlich zu den Knotenendpunkten verfügt der Redis-Cluster selbst über einen Endpunkt, den Konfigurationsendpunkt. Ihre Anwendung kann diesen Endpunkt verwenden, um aus dem Cluster zu lesen oder in den Cluster zu schreiben, wobei die Bestimmung des Knotens, von dem gelesen oder in den geschrieben werden soll, ElastiCache für Redis beibehalten wird.



Weitere Informationen finden Sie unter [Verwalten von Clustern](#).

Voraussetzungen für Skalierung

Alle Cluster können hochskaliert werden, indem ein neuer Cluster mit einem neuen, größeren Knotentyp erstellt wird. Wenn Sie einen Redis-Cluster skalieren, können Sie ihn aus einem Backup starten und vermeiden, dass der neue Cluster leer ist.

Weitere Informationen finden Sie unter [Skalierung ElastiCache für Redis](#) in diesem Handbuch.

Zugriffsvoraussetzungen

Standardmäßig wird von Amazon-EC2 Amazon EC2 ElastiCache Cluster zugegriffen. Der Netzwerkzugriff auf einen - ElastiCache Cluster ist auf das Konto beschränkt, das den Cluster erstellt hat. Bevor Sie also von einer Amazon-EC2-Instance auf einen Cluster zugreifen können, müssen Sie die Amazon-EC2-Instance für den Zugriff auf den Cluster autorisieren. Die Schritte hierzu variieren je nachdem, ob Sie ihn in EC2-VPC oder EC2-Classic starten.

Wenn Sie Ihren Cluster in EC2-VPC gestartet haben, müssen Sie Netzwerkzugang zum Cluster gewähren. Wenn Sie Ihren Cluster in EC2-Classic gestartet haben, müssen Sie der Amazon Elastic Compute Cloud-Sicherheitsgruppe, die der Instance zugeordnet ist, Zugriff auf Ihre ElastiCache Sicherheitsgruppe gewähren. Detaillierte Anweisungen finden Sie unter [Schritt 3: Zugriff auf den Cluster autorisieren](#) in diesem Handbuch.

Anforderungen für Region und Availability Zone

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 Availability Zones](#)
- [Verwenden von Local Zones mit ElastiCache](#)
- [Minimieren von Ausfällen](#)

Auswahl der Knotengröße

Die für den Cluster ausgewählte Knotengröße beeinflusst Kosten, Leistung und Fehlertoleranz.

Auswahl der Knotengröße

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 bestimmen, den Sie für Ihre Redis-Implementierung benötigen:

- Erwarten Sie durchsatzgebundene Workloads mit mehreren Client-Verbindungen?

Wenn dies der Fall ist und Sie Redis Version 5.0.6 oder höher ausführen, können Sie mit unserer erweiterten E/A-Funktion, bei der verfügbare CPUs für das Auslagern der Client-Verbindungen verwendet werden, zugunsten der Redis-Engine einen besseren Durchsatz und eine bessere Latenz erzielen. Wenn Sie Redis Version 7.0.4 oder höher zusätzlich zur erweiterten E/A-Funktion ausführen, erhalten Sie zusätzliche Beschleunigung durch erweitertes E/A-Multiplexing, bei dem jeder dedizierte Netzwerk-E/A-Thread Befehle von mehreren Clients an die Redis-Engine weiterleitet und so die Fähigkeit von Redis nutzt, Befehle effizient stapelweise zu verarbeiten. Bei ElastiCache für Redis v7.1 haben wir die erweiterte E/A-Thread-Funktionalität auf die Logik der Präsentationsebene ausgedehnt. Mit Präsentationsschicht sind die erweiterten E/A-Threads gemeint, die jetzt nicht nur Client-Eingaben lesen, sondern die Eingabe auch im Redis-Binärbefehlsformat analysieren. Diese wird dann zur Ausführung an den Haupt-Thread weitergeleitet, 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 mit Redis-Engine Version 6.2 oder höher arbeiten, können Sie Daten-Tiering nutzen, indem Sie den R6gd-Knotentyp auswählen. Bei Verwendung von Daten-Tiering werden zuletzt verwendete Daten auf dem SSD gespeichert. Beim Abruf entsteht eine geringe Latenzzeit, die jedoch durch Kosteneinsparungen ausgeglichen wird. Weitere Informationen finden Sie unter [Daten-Tiering](#).

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 Redis-Version führen Sie aus?

Bei älteren Redis-Versionen als 2.8.22 müssen Sie mehr Speicher für Failover, Snapshot, Synchronisieren und Weiterleiten eines Replikats an primäre Operationen reservieren. Diese Anforderung besteht, weil ausreichend Speicher für alle Schreibvorgänge während des Prozesses zur Verfügung stehen muss.

Redis Version 2.8.22 und höher verwenden einen unvergabelten Speicherprozess, der weniger verfügbaren Speicher erfordert als der frühere Prozess.

Weitere Informationen finden Sie hier:

- [So werden Synchronisation und Backup implementiert](#)
 - [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen 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 sind die Erstellung eines Snapshots, die Synchronisierung eines primären Clusters mit einem Replikat in einem Cluster und die Aktivierung der Append-Only-Datei-Funktion (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 Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#).

- Handelt es sich bei Ihrer Implementierung um einen eigenständigen Redis-Cluster (Cluster-Modus deaktiviert) oder um einen Redis-Cluster (Cluster-Modus aktiviert) mit mehreren Shards?

Redis-Cluster (Cluster-Modus deaktiviert)

Wenn Sie einen Redis-Cluster (Cluster-Modus deaktiviert) implementieren, muss Ihr Knotentyp in der Lage sein, all Ihre Daten plus den notwendigen Overhead, wie im vorherigen Punkt beschrieben, unterzubringen.

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.

Redis (Cluster-Modus aktiviert) mit mehreren Shards

Wenn Sie einen Redis-Cluster (mit aktiviertem Cluster-Modus) mit mehreren Shards implementieren, muss der Knotentyp in der Lage sein, `bytes-for-data-and-overhead / number-of-shards`-Bytes von Daten aufzunehmen.

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 z. B. 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 Metriken für Speichernutzung, Prozessorauslastung, Cache-Hits und Cache-Misses überwachen, die in CloudWatch veröffentlicht werden. Sie werden vielleicht feststellen, dass Ihr Cluster nicht die gewünschte Trefferquote hat oder dass die Schlüssel zu oft entfernt werden. Sie können in diesen Fällen eine andere Knotengröße mit größeren CPU- und Speicherspezifikationen wählen.

Wenn Sie die CPU-Auslastung überwachen, denken Sie daran, dass Redis ein Single-Thread-System ist. Multiplizieren Sie also die gemeldete CPU-Nutzung mit der Anzahl der CPU-Kerne, um die tatsächliche Nutzung zu erhalten. Zum Beispiel ist eine Vier-Kern-CPU, die eine 20-prozentige Auslastung meldet, in Wirklichkeit der eine Kern, der Redis mit 80 Prozent Auslastung betreibt.

Erstellen eines Clusters

Die folgenden Beispiele zeigen, wie Sie einen Redis ElastiCache -Cluster mit der AWS Management Console, AWS CLI und API erstellen.

Einen Redis erstellen (Cluster-Modus deaktiviert) (Konsole)

ElastiCache unterstützt die Replikation, wenn Sie die Redis-Engine verwenden. Um die Latenz zwischen dem Schreiben von Daten in einen primären Redis-Cluster mit Lese-/Schreibzugriff 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 Redis-Cluster vielleicht zu einem späteren Zeitpunkt zu einer Replikationsgruppe hinzugefügt wird, ist dieser Schlüssel in allen Redis-Clustern enthalten, auch wenn sie anfänglich nicht Mitglieder einer Replikationsgruppe sind. Weitere Informationen zu Replikationsgruppen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Befolgen Sie zum Erstellen eines Redis-Clusters (Cluster-Modus deaktiviert) die Schritte unter [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Sobald der Status Ihres Clusters verfügbar ist, können Sie Amazon EC2 Zugriff darauf gewähren, sich mit ihm verbinden und ihn nutzen. Weitere Informationen finden Sie unter [Schritt 3: Zugriff auf den Cluster autorisieren](#) und [Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten](#).

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](#).

Erstellen eines Redis-Clusters (Cluster-Modus aktiviert) (Konsole)

Wenn Sie Redis 3.2.4 oder höher einsetzen, können Sie einen Redis-Cluster (Cluster-Modus aktiviert) erstellen. Redis-Cluster (Cluster-Modus aktiviert) unterstützen die Partitionierung Ihrer Daten über 1 bis 500 Shards (API/CLI: Knotengruppen), allerdings mit einigen Einschränkungen. Für einen Vergleich von Redis (Cluster-Modus deaktiviert) und Redis (Cluster-Modus aktiviert) siehe [Unterstützte ElastiCache-for-Redis-Versionen](#).

So erstellen Sie einen Redis-Cluster (Cluster-Modus aktiviert) mithilfe der ElastiCache Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon- 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 diesen Cluster starten möchten.
3. Wählen Sie im Navigationsbereich Get started (Erste Schritte) aus.
4. Wählen Sie Create VPC (VPC erstellen) aus und befolgen Sie die Schritte unter [Erstellen einer Virtual Private Cloud \(VPC\)](#).
5. Wählen Sie auf der ElastiCache Dashboard-Seite Cluster erstellen und dann Redis-Cluster erstellen 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 [Minimieren von Ausfallzeiten in ElastiCache für Redis 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 [Redis-spezifische Parameter](#) und [Erstellen einer Parametergruppe](#).

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](#).

- e. Wählen Sie unter Anzahl der Shards die Anzahl der Shards (Partitionen/Knotengruppen), die Sie für diesen Redis-Cluster (Cluster-Modus aktiviert) wünschen.

Für einige Versionen von Redis (Cluster-Modus aktiviert), können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern:

- Redis 3.2.10 und höher – Wenn Ihr Cluster mit Redis 3.2.10 oder höheren Versionen arbeitet, können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern. Weitere Informationen finden Sie unter [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#).
 - Andere Redis-Versionen – Wenn Ihr Cluster eine Version von Redis vor Version 3.2.10 verwendet, gibt es einen anderen Ansatz. 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.

Für Redis (Cluster-Modus aktiviert) gelten die folgenden Einschränkungen.

- 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 [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 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, dem sowohl IPv4- als auch IPv6-Adressen zugewiesen sind, um im Dual-Stack-Modus zu arbeiten, und ein IPv6-onlySubnetz, um als IPv6-onlySubnetz zu arbeiten.

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

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 hier:

- [Auswählen eines Netzwerktyps](#).
- [Erstellen eines Subnetzes in der VPC](#)

Wenn Sie [Verwenden von Local Zones 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 aus.
- 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 Availability Zones](#).

5. Wählen Sie Next (Weiter).
6. Unter Advanced Redis settings (Erweiterte Redis-Einstellungen):
 - 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 bereitzustellen, indem Sie Kundenverwalteter AWS KMS-Schlüssel und den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwendung von kundenverwalteten Schlüsseln aus 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 bei Redis-Engine Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Optionen für die Zugriffskontrolle 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 [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
- Redis AUTH Standardbenutzer - Ein Authentifizierungsmechanismus für den Redis-Server. Weitere Informationen finden Sie unter [Redis AUTH](#).
- Redis AUTH – Ein Authentifizierungsmechanismus für Redis-Server. Weitere Informationen finden Sie unter [Redis AUTH](#).

 Note

Für Redis-Versionen ab 3.2.6, ausgenommen Version 3.2.10, ist Redis AUTH 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 Ihre VPC](#) im Benutzerhandbuch zu Amazon VPC.

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 zu Redis-Sicherungen und -Wiederherstellungen finden Sie unter [Snapshot und Wiederherstellung](#).

8. (Optional) Geben Sie ein Wartungsfenster an. Das Maintenance window (Wartungsfenster) ist der (in der Regel eine Stunde lange) allwöchentliche Zeitraum, für den ElastiCache die Systemwartung Ihres Clusters plant. Sie können es ElastiCache überlassen, einen Tag und eine Uhrzeit für das Wartungsfenster auszuwählen (No preference (Keine Präferenz)) oder Sie können den Tag, die Uhrzeit und die Dauer selbst

wählen (Specify maintenance window (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 Uhrzeiten sind in UCT angegeben.

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

9. (Optional) Für Protokolle:
 - Unter Protokollformat wählen Sie entweder Text oder JSON aus.
 - Wählen Sie unter Zieltyp entweder CloudWatch Protokolle oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen und geben Sie entweder Ihren CloudWatch Protokollgruppennamen oder Ihren Firehose-Streamnamen ein oder wählen Sie Vorhanden auswählen und wählen Sie dann entweder Ihren CloudWatch Protokollgruppennamen oder Ihren Firehose-Streamnamen aus.
10. Um Sie bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen zu unterstützen, können Sie für Tags jeder Ressource eigene Metadaten in Form von Tags zuweisen. Weitere Information finden Sie unter [Markieren Ihrer ElastiCache-Ressourcen](#).
11. Wählen Sie Weiter aus.
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 [Minimieren von Ausfallzeiten in ElastiCache für Redis mit Multi-AZ](#)
2. Befolgen Sie die Schritte unter [Verwenden von Outposts](#).

Informationen zum Erstellen des Äquivalents mithilfe der ElastiCache API oder AWS CLI anstelle der ElastiCache Konsole finden Sie im Folgenden:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Sobald als Status des Clusters available erscheint, können Sie EC2 Zugriff darauf erteilen, eine Verbindung mit ihm herstellen und ihn verwenden. Weitere Informationen finden Sie unter [Schritt](#)

[3: Zugriff auf den Cluster autorisieren](#) und [Schritt 4: Herstellen einer Verbindung zu einem Cluster-Knoten](#).

 **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](#).

Erstellen eines Clusters (AWS CLI)

Verwenden Sie den Befehl AWS CLI, um einen Cluster mit der zu erstellen `create-cache-cluster`.

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](#).

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

Example – Ein Redis-Cluster (Cluster-Modus deaktiviert) ohne Lesereplikate

Der folgende CLI-Code erstellt einen Redis-Cache-Cluster (Cluster-Modus 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 Redis-Clusters (Cluster-Modus aktiviert) (AWS CLI)

Redis-Cluster (Cluster-Modus aktiviert) (API/CLI: Replikationsgruppen) können mit dem `create-cache-cluster`-Vorgang nicht erstellt werden. Um einen Redis-Cluster (Cluster-Modus aktiviert) zu erstellen (API/CLI: Replikationsgruppe), siehe [Erstellen einer Redis-Replikationsgruppe \(Clustermodus aktiviert\) von Grund auf \(AWS CLI\)](#).

Weitere Informationen finden Sie im AWS CLI - ElastiCache Referenzthema [create-replication-group](#).

Erstellen eines Clusters (ElastiCache API)

Um einen Cluster mit der ElastiCache -API zu erstellen, verwenden Sie die `-CreateCacheCluster`Aktion.

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](#).

Themen

- [Erstellen eines Redis-Cache-Clusters \(Cluster-Modus deaktiviert\) \(ElastiCache API\)](#)
- [Erstellen eines Cache-Clusters in Redis \(Cluster-Modus aktiviert\) \(ElastiCache API\)](#)

Erstellen eines Redis-Cache-Clusters (Cluster-Modus deaktiviert) (ElastiCache API)

Der folgende Code erstellt einen Redis-Cache-Cluster (Cluster-Modus 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 Redis (Cluster-Modus aktiviert) (ElastiCache API)

Redis-Cluster (Cluster-Modus aktiviert) (API/CLI: Replikationsgruppen) können mit dem `CreateCacheCluster`-Vorgang nicht erstellt werden. Um einen Redis-Cluster (Cluster-Modus aktiviert) zu erstellen (API/CLI: Replikationsgruppe), siehe [Eine Replikationsgruppe in Redis \(Clustermodus aktiviert\) von Grund auf neu erstellen \(ElastiCache API\)](#).

Weitere Informationen finden Sie im API ElastiCache -Referenzthema [CreateReplicationGroup](#).

Anzeigen der Details eines Clusters

Sie können Detailinformationen zu einem oder mehreren Clustern über die ElastiCache-Konsole, AWS CLI oder die ElastiCache-API anzeigen.

Anzeigen von Details eines Redis-Clusters (Cluster-Modus deaktiviert) (Konsole)

Sie können die Details eines Redis-Clusters (Cluster-Modus deaktiviert) über die ElastiCache-Konsole, die AWS CLI für ElastiCache oder die ElastiCache-API anzeigen.

Im Folgenden wird beschrieben, wie Sie die Details eines Redis-Clusters (Cluster-Modus deaktiviert) mit der ElastiCache-Konsole anzeigen.

So zeigen Sie die Details eines Redis-Clusters (Cluster-Modus deaktiviert) an

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Dashboard der ElastiCache-Konsole die Option Redis, um eine Liste aller Ihrer Cluster anzuzeigen, die eine beliebige Version von Redis verwenden.
3. Markieren Sie das Kontrollkästchen links neben dem Cluster-Namen, um die Details zum Cluster anzuzeigen. Stellen Sie sicher, dass Sie einen Cluster auswählen, in dem die Redis-Engine und nicht Clustered Redis läuft. 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 mithilfe von CloudWatch-Metriken](#).
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 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 [Markieren Ihrer ElastiCache-Ressourcen](#).

Anzeigen von Details für einen Redis-Cluster (Cluster-Modus aktiviert) (Konsole)

Sie können die Details eines Redis-Clusters (Cluster-Modus aktiviert) über die ElastiCache-Konsole, die AWS CLI für ElastiCache oder die ElastiCache-API anzeigen.

Nachfolgend wird beschrieben, wie Sie die Details eines Redis-Clusters (Cluster-Modus aktiviert) über die ElastiCache-Konsole anzeigen.

Anzeigen der Details eines Redis-Clusters (Cluster-Modus aktiviert)

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS-Region aus, die Sie interessiert.
3. Wählen Sie im Dashboard der ElastiCache-Konsole die Option Redis, um eine Liste aller Ihrer Cluster anzuzeigen, die eine beliebige Version von Redis verwenden.
4. Um die Details eines Redis-Clusters (Cluster-Modus aktiviert) zu sehen, wählen Sie das Kästchen links neben dem Namen des Clusters. Stellen Sie sicher, dass Sie einen Cluster auswählen, der die Clustered Redis-Engine ausführt und nicht nur Redis.

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 mithilfe von CloudWatch-Metriken](#).
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 Network and security (Netzwerk und Sicherheit) aus, um Details zur Konfiguration von Netzwerkkonnektivität und Subnetzgruppen des Clusters, zur VPC-Sicherheitsgruppe sowie welche Verschlüsselungsmethode für den Cluster aktiviert ist, sofern zutreffend. Weitere Informationen finden Sie unter [Subnetze und Subnetzgruppen](#) und [Datensicherheit in 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 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 [Markieren Ihrer ElastiCache-Ressourcen](#).

Anzeigen der Details eines Clusters (AWS CLI)

Der folgende Code listet die Details für *my-cluster* auf:

```
aws elasticache describe-cache-clusters --cache-cluster-id my-cluster
```

Ersetzen Sie *my-cluster* in einem Fall, in dem der Cluster mit dem `create-cache-cluster`-Befehl mit 1 Cache-Knoten und 0 Shards erstellt wird, durch den Namen Ihres Clusters.

```
{
  "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 mit der AWS Management Console (Clusterknoten aktiviert oder deaktiviert, mit mindestens einem Shard) erstellt wird, verwenden Sie den folgenden Befehl, um die Details des Clusters zu beschreiben (ersetzen Sie *my-cluster* durch den Namen der Replikationsgruppe (Name Ihres Clusters)):

```
aws elasticache describe-replication-groups --replication-group-id my-cluster
```

Weitere Informationen finden Sie unter AWS CLI für das ElastiCache-Thema [describe-cache-clusters](#).

Anzeigen der Clusterdetails (ElastiCache-API)

Sie können die Details eines Clusters mit der ElastiCache-API-Aktion `DescribeCacheClusters` 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 Referenzthema [DescribeCacheClusters](#) zur ElastiCache-API .

Einen ElastiCache Cluster ändern

Zusätzlich zu dem Hinzufügen oder Entfernen von Knoten aus einem Cluster gibt es manchmal möglicherweise andere Änderungen, die Sie an einem vorhandenen Cluster vornehmen möchten, wie z. B. Hinzufügen einer Sicherheitsgruppe, 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. Anhand des Abschnitts [Änderungen treten in Kraft](#) der Spalte Details in den Tabellen für [Redis-spezifische Parameter](#) können Sie feststellen, wann eine bestimmte Parameteränderung übernommen wird.

Mit dem 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 Redis-Clients sowohl über den aktivierten als auch den deaktivierten Clustermodus eine Verbindung herstellen. Nachdem Sie alle Redis-Clients zum aktivierten Clustermodus migriert haben, können Sie die Konfiguration des Clustermodus abschließen und den Clustermodus auf Aktiviert festlegen.

- 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 [Engine-Versionen und Upgrades](#). Ein Downgrade auf ältere Engine-Versionen ist jedoch nur durch Löschen des vorhandenen Clusters und Erstellen eines neuen Clusters möglich.

- VPC Security group(s) (VPC-Sicherheitsgruppe(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 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 am Knotentyp und an der Engine-Version während des nächsten Wartungsfensters übernommen. Andere Änderungen, wie z. B. Ändern des Wartungsfensters, werden umgehend übernommen.

7. Wählen Sie Ändern aus.

Aktivieren/Deaktivieren Sie die Protokollzustellung wie folgt:

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 Cluster details (Cluster-Details) die Registerkarte Logs (Protokolle) aus.
3. Um langsame Protokolle zu aktivieren oder zu deaktivieren, wählen Sie entweder Enable (Aktivieren) oder Disable (Deaktivieren).

Wenn Sie „Aktivieren“ wählen:

- a. Wählen Sie unter Log format (Protokollformat) entweder JSON oder Text aus.
- b. Wählen Sie unter Protokollzieltyp entweder CloudWatch Logs oder Kinesis Firehose aus.
- c. 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 Loggruppennamen oder Ihren Kinesis Data Firehose Firehose-Stream-Namen.
- d. Wählen Sie Enable (Aktivieren) aus.

Zur Änderung Ihrer Konfiguration:

1. Wählen Sie Modify (Ändern) aus.
2. Wählen Sie unter Log format (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 den AWS CLI

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 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 [Engine-Versionen und Upgrades](#). 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 AWS CLI ElastiCache For-Thema [modify-cache-cluster](#).

Verwenden der ElastiCache API

Sie können einen vorhandenen Cluster mithilfe der ElastiCache `ModifyCacheCluster` API-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 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 [Engine-Versionen und Upgrades](#). 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 ElastiCache API-Referenzthema [ModifyCacheCluster](#).

Knoten zu einem Cluster hinzufügen

Um Ihren Redis-Cluster (Cluster-Modus aktiviert) neu zu konfigurieren, siehe [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#)

Sie können die ElastiCache-Managementkonsole, die AWS CLI oder die ElastiCache-API verwenden, um Knoten zu Ihrem Cluster hinzuzufügen.

Verwendung von AWS Management Console

Wenn Sie einen Knoten zu einem Redis-Cluster mit nur einem Knoten (Cluster-Modus deaktiviert) (ohne aktivierte Replikation) hinzufügen möchten, ist dies ein zweistufiger Prozess: Fügen Sie zuerst die Replikation und dann einen Replikationsknoten hinzu.

Themen

- [So fügen Sie zu einem Redis-Cluster ohne Shards die Replikation hinzu](#)
- [So fügen Sie Knoten zu einem Cluster hinzu \(Konsole\)](#)

Das folgende Verfahren fügt einen Redis-Cluster mit einem einzelnen Knoten hinzu, bei dem keine Replikation 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.

So fügen Sie zu einem Redis-Cluster ohne Shards die Replikation hinzu

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

Es wird eine Liste der Cluster der Redis-Engine angezeigt.

3. Wählen Sie den Namen eines Clusters, nicht das Kästchen links vom Namen des Clusters, dem Sie Knoten hinzufügen möchten.

Folgendes trifft auf einen Redis-Cluster zu, bei dem keine Replikation aktiviert ist:

- Er führt Redis und nicht Clustered Redis aus.
- Er besitzt keine Shards.

Wenn der Cluster über Shards verfügt, ist die Replikation für ihn bereits aktiviert und Sie können mit [So fügen Sie Knoten zu einem Cluster hinzu \(Konsole\)](#) fortfahren.

4. Wählen Sie Add replication.
5. Geben Sie unter Add Replication eine Beschreibung für diesen replikationsfähigen Cluster ein.
6. Wählen Sie Add (Hinzufügen) aus.

Sobald der Status des Clusters wieder available lautet, können Sie mit dem nächsten Verfahren fortfahren und Replikate zum Cluster hinzufügen.

So fügen Sie Knoten zu einem Cluster hinzu (Konsole)

Die folgende Vorgehensweise kann verwendet werden, um Knoten zu einem Cluster hinzuzufügen.

1. Melden Sie sich bei AWS Management Console an 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 Ihr Cluster ein Redis-Cluster (Cluster-Modus aktiviert) ist, siehe [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#).

Wenn Ihr Cluster ein Redis-Cluster (Cluster-Modus deaktiviert) mit null Shards ist, komplettieren Sie zunächst die Schritte unter [So fügen Sie zu einem Redis-Cluster ohne Shards die Replikation hinzu](#).

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> |

| Szenarien | Anstehende Operation | Neue Anfrage | Ergebnisse |
|------------|----------------------|--------------|---|
| 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.

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.

Verwendung von AWS CLI

Wenn Sie einem vorhandenen Redis-Cluster (Cluster-Modus deaktiviert), in dem die Replikation nicht aktiviert ist, Knoten hinzufügen möchten, müssen Sie zunächst die Replikationsgruppe erstellen und dabei den vorhandenen Cluster als primär angeben. Weitere Informationen finden Sie unter [Erstellen einer Replikationsgruppe mit einem verfügbaren Redis-Cache-Cluster \(AWS CLI\)](#). Wenn die Replikationsgruppe verfügbar ist, können Sie mit dem folgenden Vorgang fortfahren.

Um dem Cluster mithilfe der AWS CLI Knoten hinzuzufügen, verwenden Sie die AWS CLI-Operation `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 ähnlich der folgenden (JSON-Format):

```
{  
  "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](#).

Verwenden der ElastiCache-API

Wenn Sie einem vorhandenen Redis-Cluster (Cluster-Modus deaktiviert), in dem die Replikation nicht aktiviert ist, Knoten hinzufügen möchten, müssen Sie zunächst die Replikationsgruppe erstellen und dabei den vorhandenen Cluster als primär angeben. Weitere Informationen finden Sie unter [Hinzufügen von Replikaten zu einem eigenständigen Redis-Cluster \(Cluster-Modus deaktiviert\) \(ElastiCache API\)](#). Wenn die Replikationsgruppe verfügbar ist, können Sie mit dem folgenden Vorgang fortfahren.

Hinzufügen von Knoten zu einem Cluster (ElastiCache API)

- Rufen Sie die API-Operation `IncreaseReplicaCount` mit 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 kleiner ist als die aktuelle Anzahl der Knoten, verwenden Sie die `DecreaseReplicaCount`-API 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, dem 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 im Thema ElastiCache-API [IncreaseReplicaCount](#).

Entfernen von Knoten aus einem Cluster

Sie können einen Knoten aus einem Cluster mit der AWS Management Console, der AWS CLI oder der ElastiCache-API löschen.

Verwendung von AWS Management Console

Entfernen von Knoten aus einem Cluster (Konsole)

1. Melden Sie sich bei AWS Management Console an 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 der 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. In 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 das Löschen des Knotens dazu führt, dass der Cluster nicht mehr Multi-AZ-konform ist, deaktivieren Sie zuerst das Kontrollkästchen Multi-AZ und löschen Sie dann den Knoten. 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 gestellt, werden keine neuen Knoten angelegt und der Knoten 0003 wird gelöscht.</p> |
| 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</p> |

| Szenarien | Anstehende Operation | Neue Anfrage | Ergebnisse |
|-----------|----------------------|--------------|---|
| | | | <p>Anforderung hinzugefügt und es werden fünf Knoten erstellt.</p> <div style="border: 1px solid #f08080; 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.


Verwendung von AWS CLI

1. Identifizieren Sie die IDs der Knoten, die Sie entfernen möchten. Weitere Informationen finden Sie unter [Anzeigen der Details eines Clusters](#).
2. Listen Sie mit der CLI-Operation `decrease-replica-count` wie im folgenden Beispiel die zu entfernenden Knoten auf.

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 Knoten-IDs, 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 der 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 ähnlich der folgenden (JSON-Format):

```
{  
  "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 der ElastiCache-API

Um Knoten mit der ElastiCache-API zu entfernen, rufen Sie den `DecreaseReplicaCount`-API-Vorgang mit der Replikationsgruppen-ID und einer Liste der zu entfernenden Knoten auf wie 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 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=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 im Thema ElastiCache-API [DecreaseReplicaCount](#).

Abbrechen ausstehender Vorgänge zum Hinzufügen oder Löschen von Knoten

Wenn Sie ausgewählt haben, dass eine Änderung nicht umgehend übernommen wird, ist der Status der Operation pending, bis sie im nächsten Wartungsfenster ausgeführt wird. Anstehende Operationen können storniert werden.

So stornieren Sie eine anstehende Operation

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Aus der Liste in der oberen rechten Ecke wählen Sie die AWS-Region, 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

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-for-Redis-Cluster löschen, bleiben Ihre manuellen Snapshots erhalten. Sie können auch einen letzten Snapshot erstellen, bevor der Cluster gelöscht wird. Automatisch erstellte Cache-Snapshots werden nicht aufbewahrt.

Verwendung von 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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Dashboard der ElastiCache-Konsole die Engine aus, in der der zu löschende Cluster läuft.

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 immer nur einen Cluster auf einmal 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.

Verwendung von AWS CLI

Der folgende Code löscht den Cache-Cluster `my-cluster`.

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

Die CLI-Aktion `delete-cache-cluster` 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 unter AWS CLI für das ElastiCache-Thema [delete-cache-cluster](#).

Verwenden der 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>
```

Die API-Operation `DeleteCacheCluster` 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](#).

Zugriff auf Ihren Cluster oder die Replikationsgruppe

Ihre Amazon-ElastiCache-Instances sind für den Zugriff über eine Amazon-EC2-Instance konzipiert.

Wenn Sie Ihre ElastiCache-Instance in einer Amazon Virtual Private Cloud (Amazon VPC) gestartet haben, können Sie von einer Amazon-EC2-Instance in derselben Amazon VPC auf Ihre ElastiCache-Instance zugreifen. Oder Sie können mit Hilfe von VPC-Peering auf Ihre ElastiCache-Instance von einer Amazon EC2 in einer anderen Amazon VPC 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 verbundenen Amazon-EC2-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 in EC2-VPC gestartet

Falls Sie Ihren Cluster in einer Amazon Virtual Private Cloud (Amazon VPC) gestartet haben, können Sie sich nur von einer Amazon-EC2-Instance, die in derselben Amazon VPC läuft, mit Ihrem ElastiCache-Cluster verbinden. 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](#). Dadurch wird Ihre VPC auf diese Local Zone ausgedehnt, und Ihre VPC behandelt das Subnetz wie jedes andere Subnetz in jeder anderen Availability Zone, und relevante Gateways, Routentabellen und andere Sicherheitsgruppen werden automatisch angepasst.

So gewähren Sie einem Cluster den Netzwerkeingang aus einer Amazon-VPC-Sicherheitsgruppe

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-EC2-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 Ihrer Amazon VPC aus. Sofern Sie keine Sicherheitsgruppe für die Verwendung von ElastiCache erstellt haben, wird diese Sicherheitsgruppe als Standard bezeichnet.
4. Wählen Sie die Registerkarte Inbound und verfahren Sie dann wie folgt:
 - a. Wählen Sie Bearbeiten aus.
 - b. Wählen Sie Regel hinzufügen aus.
 - c. Wählen Sie in der Spalte Type die Option Custom TCP Rule 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 Redis ist **6379**.
 - e. Wählen Sie im Feld Quelle die Option Überall mit dem Portbereich (0.0.0.0/0), damit jede Amazon-EC2-Instance, die Sie innerhalb Ihrer Amazon VPC starten, eine Verbindung zu Ihren ElastiCache-Knoten herstellen kann.

 **Important**

Durch die Öffnung des ElastiCache-Clusters auf 0.0.0.0/0 wird der Cluster nicht dem Internet ausgesetzt, da er keine öffentliche IP-Adresse hat und daher von außerhalb der VPC nicht erreichbar ist. Die Standardsicherheitsgruppe kann jedoch auf andere Amazon-EC2-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 ElastiCache verwendet wird. Weitere Informationen finden Sie unter [Benutzerdefinierte Sicherheitsgruppen](#).

- f. Wählen Sie Speichern.

Wenn Sie eine Amazon-EC2-Instance in Ihrer Amazon VPC starten, kann sich diese Instance mit Ihrem ElastiCache-Cluster verbinden.

Zugriff auf ElastiCache-Ressourcen außerhalb von AWS

Amazon ElastiCache ist ein AWS-Service, der einen cloudbasierten In-Memory-Schlüssel-Wert-Speicher bereitstellt. Der Service ist für den exklusiven Zugriff innerhalb von AWS ausgelegt. Wenn der ElastiCache-Cluster jedoch in einer VPC gehostet wird, können Sie eine Network Address Translation (NAT)-Instance verwenden, um den Zugriff von außen zu ermöglichen.

Voraussetzungen

Die folgenden Anforderungen müssen erfüllt sein, damit Sie von außerhalb von AWS auf Ihre ElastiCache-Ressourcen zugreifen können:

- Der Cluster muss sich innerhalb einer VPC befinden und über eine Network Address Translation (NAT)-Instance zugänglich sein. Für diese Anforderung gibt es keine Ausnahmen.
- Die NAT-Instance muss in der gleichen VPC gestartet werden wie der Cluster.
- Die NAT-Instance muss in einem vom Cluster getrennten, öffentlichen Subnetz gestartet werden.
- Eine Elastic IP-Adresse (EIP) muss der NAT-Instance zugeordnet sein. Die Portweiterleitungsfunktion von iptables wird verwendet, um einen Port auf der NAT-Instance an den Cache-Knotenport innerhalb der VPC weiterzuleiten.

Überlegungen

Beachten Sie beim Zugriff auf Ihre ElastiCache-Ressourcen von außerhalb von ElastiCache die folgenden Punkte.

- Clients stellen eine Verbindung mit der EIP und dem 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-Instance fungiert als Proxy zwischen Clients und mehreren Clustern. Ein zusätzlicher Proxy beeinträchtigt die Leistung des Cache-Clusters. Die Beeinträchtigung steigt mit der Anzahl von Cache-Clustern, auf die Sie über die NAT-Instance zugreifen.
- Der Datenverkehr von den Clients zur NAT-Instance ist nicht verschlüsselt. Daher sollten Sie keine sensiblen Daten über die NAT-Instance senden.
- Die NAT-Instance bedeutet zusätzlichen Aufwand durch Verwalten einer weiteren Instance.
- Die NAT-Instance dient als einzelne Fehlerquelle. Informationen zum Einrichten von NAT mit hoher Verfügbarkeit in VPC finden Sie unter [Hohe Verfügbarkeit für Amazon VPC NAT-Instances: Beispiel](#).

So greifen Sie auf ElastiCache-Ressourcen außerhalb von AWS zu

Im Folgenden wird erläutert, wie Sie eine Verbindung mit Ihren ElastiCache-Ressourcen mithilfe einer NAT-Instance herstellen können.

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 Redis-Cluster erfolgt mit:
 - IP-Adresse – 10.0.1.230
 - Redis-Standardport – 6379
 - Sicherheitsgruppe – sg-bd56b7da
 - IP-Adresse der AWS-Instance – 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.
- Die Sicherheitsgruppe Ihrer NAT-Instance lautet sg-ce56b7a9.

So stellen Sie eine Verbindung mit Ihren ElastiCache-Ressourcen mithilfe einer NAT-Instance her

1. Erstellen Sie eine NAT-Instance in derselben VPC wie Ihr Cache-Cluster, allerdings in einem öffentlichen Subnetz.

Standardmäßig startet der VPC-Assistent einen cache.m1.small-Knotentyp. Wählen Sie eine Knotengröße aus, die Ihren Anforderungen entspricht. Sie müssen das EC2-NAT-AMI verwenden, um außerhalb von AWS auf ElastiCache zugreifen zu können.

Weitere Informationen zum Erstellen einer NAT-Instance finden Sie unter [NAT-Instances](#) im AWS-VPC-Benutzerhandbuch.

2. Erstellen Sie Sicherheitsgruppenregeln für den Cache-Cluster und die NAT-Instance.

Die Sicherheitsgruppe der NAT-Instance und die Cluster-Instance sollten folgende Regeln besitzen:

- Zwei Regeln für eingehenden Datenverkehr
 - Eine Regel, mit der TCP-Verbindungen von vertrauenswürdigen Clients mit jedem Cache-Port zulässig sind, die von der NAT-Instance weitergeleitet wird (6379 – 6381).
 - Eine zweite Regel, die SSH-Zugriff auf vertrauenswürdige Clients zulässt.

Sicherheitsgruppe der NAT-Instance – Regeln für eingehenden Datenverkehr

| Typ | Protocol (Protokoll) | Port-Bereich | Quelle |
|-------------|----------------------|--------------|------------------|
| Zielbereich | TCP | 6379-6380 | 198.51.100.27/32 |
| SSH | TCP | 22 | 203.0.113.73/32 |

- Eine Regel für ausgehenden Datenverkehr, der TCP-Verbindungen mit jedem Cache-Port zulässt (6379).

Sicherheitsgruppe der NAT-Instance – Regeln für ausgehenden Datenverkehr

| Typ | Protocol (Protokoll) | Port-Bereich | Bestimmungsort |
|-------------|----------------------|--------------|--|
| Zielbereich | TCP | 6379 | sg-ce56b7a9 (Sicherheitsgruppe der Cluster-Instance) |

- Eine Regel für eingehenden Datenverkehr für die Sicherheitsgruppe des Clusters, die TCP-Verbindungen von der NAT-Instance mit dem Cache-Port zulässt (6379).

Sicherheitsgruppe der Cluster-Instance – Regel für eingehenden Datenverkehr

| Typ | Protocol (Protokoll) | Port-Bereich | Quelle |
|-------------|----------------------|--------------|---|
| Zielbereich | TCP | 6379 | sg-bd56b7da (Cluster-Sicherheitsgruppe) |

3. Validieren Sie die Regeln.

- Bestätigen Sie, dass der vertrauenswürdige Client eine SSH-Verbindung mit der NAT-Instance herstellen kann.
- Bestätigen Sie, dass der vertrauenswürdige Client eine Verbindung mit dem Cluster über die NAT-Instance herstellen kann.

4. Fügen Sie der NAT-Instance eine iptables-Regel hinzu.

Eine iptables-Regel muss der NAT-Tabelle für jeden Knoten im Cluster hinzugefügt werden, um den Cache-Port von der NAT-Instance an den Cluster-Knoten 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 z. B. mit einem aus drei Knoten bestehenden Redis-Cluster unter Verwendung der Ports 6379 – 6381 arbeiten, sehen die Regeln wie folgt aus:

```
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 mit dem EIP herstellen, der der NAT-Instance zugewiesen ist, und dem Cluster-Port, der dem entsprechenden Cluster-Knoten entspricht. Die Verbindungszeichenfolge für PHP sieht beispielsweise wie folgt aus:

```
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. Beispiel:

```
telnet 203.0.113.73 6379
telnet 203.0.113.73 6380
telnet 203.0.113.73 6381
```

- 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 einer Amazon VPC](#)
- [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird](#)

- [NAT-Instances](#)
- [Konfigurieren von ElastiCache-Clients](#)
- [Hohe Verfügbarkeit für Amazon VPC NAT-Instances: Beispiel](#)

Verbindungsendpunkte ermitteln

Ihre Anwendung stellt über Endpunkte eine Verbindung zu Ihrem Cluster her. Ein Endpunkt ist die eindeutige Adresse eines Knotens oder Clusters.

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.

Zu verwendende Endpunkte

- Redis-Standalone-Knoten – Verwenden Sie den Endpunkt für diesen Knoten sowohl für Lese- als auch für Schreibvorgänge.
- Redis-Cluster (Cluster-Modus deaktiviert), verwenden Sie den Primären Endpunkt für alle Schreibvorgänge. Verwenden Sie Leser-Endpunkt, um am Endpunkt ankommende Verbindungen auf alle Lesereplikate zu verteilen. Verwenden Sie die einzelnen Knotenendpunkte für Lesevorgänge (in der API/CLI werden diese als Leseendpunkte bezeichnet).
- Redis-Cluster (Cluster-Modus aktiviert), verwenden Sie den Konfigurationsendpunkt des Clusters für alle Vorgänge, die Befehle im Cluster-Modus unterstützen. Sie müssen einen Client verwenden, der Redis Cluster (Redis 3.2) unterstützt. Sie können weiterhin aus einzelnen Knotenendpunkten lesen (in der API/CLI werden diese als Leseendpunkte bezeichnet).

Die folgenden Abschnitte führen Sie durch die Erkennung der Endpunkte, die Sie für die von Ihnen ausgeführte Engine benötigen.

Suche nach den Endpunkten eines Redis-Clusters (Cluster-Modus deaktiviert) (Konsole)

Wenn ein Redis-Cluster (Cluster-Modus deaktiviert) nur einen Knoten hat, wird der Endpunkt des Knotens sowohl für Lese- als auch für Schreibvorgänge verwendet. Wenn ein Redis-Cluster (Cluster-Modus deaktiviert) mehrere Knoten hat, gibt es drei Arten von Endpunkten: den Primären Endpunkt, den Leser-Endpunkt und die Knoten-Endpunkte.

Der primäre Endpunkt ist ein DNS-Name, der immer auf den 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 wird eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Lesereplikate in einem ElastiCache-for-Redis-Cluster aufteilen. 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 Lese-Replikate Ihres ElastiCache for Redis-Clusters in verschiedenen AWS-Availability Zones (AZ) platzieren, um eine hohe Verfügbarkeit der Lese-Endpunkte zu gewährleisten.

Note

Ein Leser-Endpunkt ist kein Load Balancer. Es handelt sich um einen DNS-Eintrag, der nach dem Rotationsprinzip zu einer IP-Adresse eines der Replikationsknoten 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 Redis-Clusters (Cluster-Modus deaktiviert)

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

Der Cluster-Bildschirm erscheint mit einer Liste von Redis-Clustern (Cluster-Modus deaktiviert) und Redis-Clustern (Cluster-Modus aktiviert).

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

Primäre und Leser-Endpunkte für einen Redis-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 Redis-Cluster (Cluster-Modus deaktiviert) über Replikationsknoten verfügt, können Sie die Replikationsknoten-Endpunkte des Clusters finden, indem Sie den Namen des Clusters und die Registerkarte Nodes (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 | [redacted]amazonaws.com |
| <input type="checkbox"/> | test-no-002 | available | replica | 6379 | [redacted]amazonaws.com |
| <input type="checkbox"/> | test-no-003 | available | replica | 6379 | [redacted]amazonaws.com |

Knotenendpunkte für einen Redis-Cluster (Cluster-Modus 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 (Cluster-Modus 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
```

Suchen von Endpunkten für einen Redis-Cluster (Cluster-Modus aktiviert) (Konsole)

Ein Redis-Cluster (Cluster-Modus 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.

Finden Sie die Endpunkte für einen Redis-Cluster (Cluster-Modus aktiviert) wie folgt:

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

Der Cluster-Bildschirm erscheint mit einer Liste von Redis-Clustern (Cluster-Modus deaktiviert) und Redis-Clustern (Cluster-Modus aktiviert). Wählen Sie den Redis-Cluster (Cluster-Modus aktiviert), zu dem Sie eine Verbindung herstellen möchten.

3. Um den Konfigurationsendpunkt des Clusters zu suchen, wählen Sie den Namen des Clusters (nicht das Optionsfeld) aus.
4. Der Configuration endpoint (Konfigurationsendpunkt) wird unter Cluster details (Cluster-Details) angezeigt. Wählen Sie zum Kopieren das Symbol Copy (Kopieren) links vom Endpunkt aus.

Suchen von Endpunkten (AWS CLI)

Sie können den AWS CLI für Amazon ElastiCache verwenden, um die Endpunkte für Knoten, Cluster und Replikationsgruppen zu ermitteln.

Themen

- [Suchen von Endpunkten für Knoten und Cluster \(AWS CLI\)](#)
- [Suchen der Endpunkte für Replikationsgruppen \(AWS CLI\)](#)

Suchen von Endpunkten für Knoten und Cluster (AWS CLI)

Sie können über die AWS CLI die Endpunkte für einen Cluster und seine Knoten mithilfe des Befehls `describe-cache-clusters` ermitteln. Für Redis-Cluster gibt der Befehl den Cluster-Endpunkt 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 die Clusterinformationen für den Redis-Cluster mit einem Knoten (Cluster-Modus deaktiviert) `mycluster` ab.

Important

Der Parameter `--cache-cluster-id` kann mit der Cluster-ID eines Redis-Einzelknotens (Cluster-Modus deaktiviert) oder bestimmten Knoten-IDs in Redis-Replikationsgruppen verwendet werden. Der `--cache-cluster-id` einer Redis-Replikationsgruppe ist ein vierstelliger Wert, z. B. `0001`. Wenn `--cache-cluster-id` die ID eines Clusters (Knoten) in einer Redis-Replikationsgruppe ist, ist `replication-group-id` 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 etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{  
  "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](#).

Suchen der Endpunkte für Replikationsgruppen (AWS CLI)

Sie können über die AWS CLI die Endpunkte für eine Replikationsgruppe und ihre Cluster mithilfe des Befehls `describe-replication-groups` ermitteln. 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 dieser Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "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 unter [describe-replication-groups](#) in der AWS CLIBefehlsreferenz.

Finden von Endpunkten (ElastiCache-API)

Sie können die Amazon-ElastiCache-API verwenden, um die Endpunkte für Knoten, Cluster und Replikationsgruppen zu ermitteln.

Themen

- [Finden von Endpunkten für Knoten und Cluster \(ElastiCache-API\)](#)
- [Finden von Endpunkten für Replikationsgruppen \(ElastiCache-API\)](#)

Finden von Endpunkten für Knoten und Cluster (ElastiCache-API)

Sie können die ElastiCache-API verwenden, um die Endpunkte für einen Cluster und seine Knoten mit der `DescribeCacheClusters`-Aktion zu ermitteln. Für Redis-Cluster gibt der Befehl den Cluster-Endpoint zurück. Wenn Sie den optionalen Parameter `ShowCacheNodeInfo` einschließen, gibt die Aktion auch die Endpunkte der einzelnen Knoten im Cluster zurück.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=DescribeCacheClusters  
  &CacheClusterId=mycluster  
  &ShowCacheNodeInfo=true  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &Version=2015-02-02  
  &X-Amz-Credential=<credential>
```

Finden von Endpunkten für Replikationsgruppen (ElastiCache-API)

Sie können die ElastiCache-API verwenden, um die Endpunkte für eine Replikationsgruppe und ihre Cluster mit der `DescribeReplicationGroups`-Aktion zu ermitteln. Die Aktion gibt den primären Endpunkt der Replikationsgruppe und eine Liste aller Cluster in der Replikationsgruppe mit ihren Endpunkten sowie den Reader-Endpoint zurück.

Die folgende Operation ruft den primären Endpunkt (`PrimaryEndpoint`), den Reader-Endpoint (`ReaderEndpoint`) und die einzelnen Knotenendpunkte (`ReadEndpoint`) der Replikationsgruppe `myreplgroup` ab. 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

Ein Shard (API/CLI: Knotengruppe) ist eine Sammlung von einem bis sechs Redis-Knoten. Ein Redis-Cluster (Cluster-Modus deaktiviert) hat nie mehr als einen Shard. Mit Shards können Sie große Datenbanken in kleinere, schnellere und einfacher verwaltete Teile unterteilen, die als Daten-Shards bezeichnet werden. Dies kann die Datenbankeffizienz erhöhen, indem Vorgänge auf mehrere separate Abschnitte verteilt werden. Die Verwendung von Shards kann viele Vorteile bieten, darunter 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 Version der Redis-Engine 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. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht werden. Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#).

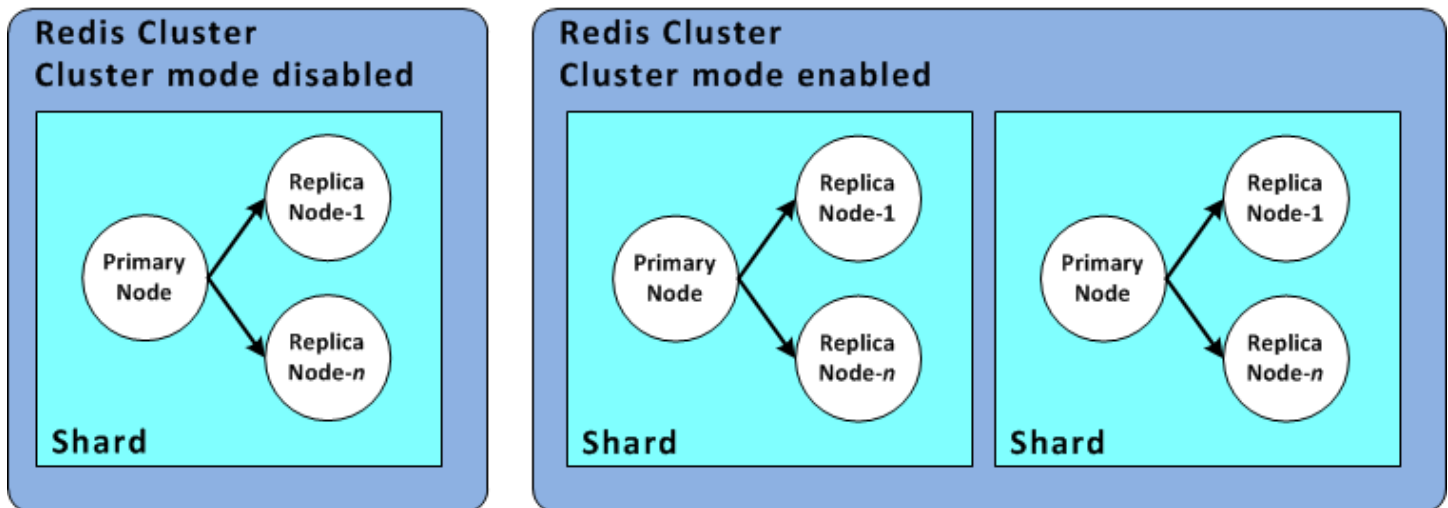
Für Versionen unter 5.0.6 liegt das Limit bei 250 pro Cluster.

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

Wenn Sie einen Redis-Cluster (Cluster-Modus aktiviert) mit der ElastiCache Konsole 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 Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#). Wenn Sie die ElastiCache -API oder verwenden AWS CLI , um einen Cluster zu erstellen (in der API/CLI als Replikationsgruppe bezeichnet), können Sie die Anzahl der Knoten in einem Shard (API/CLI: Knotengruppe) unabhängig konfigurieren. Weitere Informationen finden Sie hier:

- API: [CreateReplicationGroup](#)
- CLI: [create-replication-group](#)

Für jeden Knoten in einem Shard gelten dieselben Rechner-, Arbeitsspeicher- und Festspeicherspezifikationen. Mit der ElastiCache API können Sie Shard-weite Attribute wie die Anzahl der Knoten, Sicherheitseinstellungen und Systemwartungsfenster steuern.



Redis-Shard-Konfigurationen

Weitere Informationen finden Sie unter [Offline-Resharding und Shard-Rebalancing für Redis \(Cluster-Modus aktiviert\)](#) und [Online-Resharding und Shard-Rebalancing für Redis \(Clustermodus aktiviert\)](#).

Auffinden der ID eines Shards

Sie finden die ID eines Shards mithilfe der AWS Management Console, der AWS CLI oder der ElastiCache API.

Verwenden der AWS Management Console

Themen

- [Für Redis \(Cluster-Modus deaktiviert\)](#)
- [Für Redis \(Cluster-Modus aktiviert\)](#)

Für Redis (Cluster-Modus deaktiviert)

Redis-Replikationsgruppen-Shard-IDs (Cluster-Modus deaktiviert) sind immer 0001.

Für Redis (Cluster-Modus aktiviert)

Im folgenden Verfahren wird verwendet AWS Management Console , um die Shard-ID einer Redis-Replikationsgruppe (Cluster-Modus aktiviert) zu finden.

So finden Sie die Shard-ID in einer Redis-Replikationsgruppe (Cluster-Modus 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 Redis und dann den Namen der Redis-Replikationsgruppe (Cluster-Modus aktiviert) aus, für die Sie die Shard-IDs suchen möchten.
3. In der Spalte Shard-Name ist die Shard-ID die letzten vier Ziffern des Shard-Namens.

Verwenden der AWS CLI

Um Shard-IDs (Knotengruppe) für Redis-Replikationsgruppen (Cluster-Modus deaktiviert) oder Redis-Replikationsgruppen (Cluster-Modus aktiviert) zu finden, verwenden Sie die - AWS CLI Operation `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- bzw. Knotengruppen-IDs werden hier *hervorgehoben*, damit sie leichter auffindbar sind.

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

```

Verwenden der ElastiCache API

Um Shard-IDs (Knotengruppe) für Redis-Replikationsgruppen (Cluster-Modus deaktiviert) oder Redis-Replikationsgruppen (Cluster-Modus aktiviert) zu finden, verwenden Sie die - AWS CLI Operation `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 Memcached und Redis

Amazon ElastiCache unterstützt die Cache-Engines Memcached und Redis. 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 Redis mit einer Version von ElastiCache for Redis, wenn Folgendes auf Sie zutrifft:

- ElastiCache für Redis Version 7.0 (erweitert)

Sie sollten die [Redis-Funktionen](#), [Sharded Pub/Sub](#) oder [Redis-ACL-Verbesserungen](#) verwenden. Weitere Informationen finden Sie unter [Redis Version 7.0 \(Erweitert\)](#).

- ElastiCache für Redis Version 6.2 (erweitert)

Sie möchten die Möglichkeit haben, Daten zwischen Speicher und SSD mit dem R6gd-Knotentyp zu staffeln. Weitere Informationen finden Sie unter [Daten-Tiering](#).

- ElastiCache für Redis Version 6.0 (erweitert)

Sie möchten Benutzer mit rollenbasierter Zugriffskontrolle authentifizieren.

Weitere Informationen finden Sie unter [Redis Version 6.0 \(Erweitert\)](#).

- ElastiCache für Redis Version 5.0.0 (erweitert)

Sie möchten [Redis streams \(Redis-Streams\)](#) verwenden, eine Protokolldatenstruktur, mit der neue Elemente in Echtzeit angehängt und Nachrichten blockierend oder blockierungsfrei abgerufen werden können.

Weitere Informationen finden Sie unter [Redis-Version 5.0.0 \(Erweitert\)](#).

- ElastiCache für Redis Version 4.0.10 (erweitert)

Unterstützt sowohl die Verschlüsselung als auch das dynamische Hinzufügen oder Entfernen von Shards aus Ihrem Redis-Cluster (Cluster-Modus aktiviert).

Weitere Informationen finden Sie unter [Redis-Version 4.0.10 \(Erweitert\)](#).

Die folgenden Versionen sind veraltet, haben das Ende ihrer Lebensdauer erreicht oder stehen kurz vor dem Ende.

- ElastiCache für Redis Version 3.2.10 (erweitert)

Unterstützt die Möglichkeit, Shards dynamisch zu Ihrem Redis-Cluster (Cluster-Modus aktiviert) hinzuzufügen oder daraus zu entfernen.

⚠ Important

Derzeit unterstützt ElastiCache Redis 3.2.10 keine Verschlüsselung.

Weitere Informationen finden Sie hier:

- [Redis Version 3.2.10 \(Enhanced\)](#)
- Bewährte Methoden für Online-Resharding für Redis und weitere Informationen finden Sie unter:
 - [Bewährte Methoden: Online-Resharding](#)
 - [Online-Resharding und Shard-Rebalancing für Redis \(Cluster-Modus aktiviert\)](#)
- Weitere Informationen zur Skalierung von Redis-Clustern finden Sie unter [Skalierung](#).
- ElastiCache für Redis Version 3.2.6 (erweitert)

Wenn Sie die Funktionalität früherer Redis-Versionen sowie die folgenden Funktionen benötigen, wählen Sie ElastiCache Redis 3.2.6:

- Verschlüsselung während der Übertragung. Weitere Informationen finden Sie unter [Amazon ElastiCache for Redis In-Transit Encryption](#).
- Verschlüsselung im Ruhezustand. Weitere Informationen finden Sie unter [Amazon ElastiCache for Redis At-Rest Encryption](#).
- ElastiCache für Redis (Cluster-Modus aktiviert) Version 3.2.4

Wählen Sie Redis 3.2.4 (Cluster-Modus) aus, wenn Sie die Funktionalität von Redis 2.8.x und die folgenden Funktionen benötigen:

- 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 für Redis (nicht geclusterter Modus) 2.8.x und 3.2.4 (erweitert)

Wählen Sie Redis 2.8.x oder Redis 3.2.4 (Nicht-Cluster-Modus) aus, wenn Folgendes für Sie gilt:

- 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 Lesereplike 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, Redis (Cluster-Modus deaktiviert) und Redis (Cluster-Modus aktiviert)

| | Memcached | Redis (Cluster-Modus deaktiviert) | Redis (Cluster-Modus 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-Zertifizierung | | | |
| FedRAMP | Ja – 1.6.12 und höher | 4.0.10 und höher | 4.0.10 und höher |
| HIPAA | Ja – 1.6.12 und höher | 4.0.10 und höher | 4.0.10 und höher |
| PCI DSS | Ja | 4.0.10 und höher | 4.0.10 und höher |
| Multi-Thread | Ja | Nein | Nein |
| Upgrade des Knotentyps | Nein | Ja | Ja |
| Engine-Upgrade | Ja | Ja | Ja |

| | Memcached | Redis (Cluster-Modus deaktiviert) | Redis (Cluster-Modus aktiviert) |
|----------------------------------|--|-----------------------------------|---------------------------------|
| 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 Unterstützt ElastiCache für Memcached-Versionen oder Unterstützt für Redis-Versionen. ElastiCache](#)

Online-Migration zu ElastiCache

Mit der Online-Migration können Sie Ihre Daten von selbst gehostetem Open-Source-Redis in Amazon EC2 zu Amazon ElastiCache migrieren.


 Note

Die Online-Migration wird für ElastiCache-Serverless-Caches oder Cluster mit r6gd-Knotentyp nicht unterstützt.

Übersicht

Für die Migration Ihrer Daten von Open-Source-Redis, das auf Amazon EC2 läuft, zu Amazon ElastiCache ist eine bestehende oder neu erstellte Amazon-ElastiCache-Bereitstellung erforderlich. 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 gehostetem Open-Source-Redis in Amazon EC2 zu ElastiCache für Redis und nicht zwischen Clustern von ElastiCache für Redis vorgesehen.

 Important

Wir empfehlen dringend, die folgenden Abschnitte vollständig zu lesen, bevor Sie den Online-Migrationsprozess starten.

Die Migration beginnt, sobald Sie die API-Operation oder den AWS CLI-Befehl `StartMigration` aufrufen. Bei Redis mit aktiviertem Cluster-Modus wird der Primärknoten des Clusters von ElastiCache für Redis durch den Migrationsprozess zu einem Replikat Ihres Quell-Redis-Primärknotens. Bei Redis mit aktiviertem Cluster-Modus wird der Primärknoten jedes ElastiCache-Shards durch den Migrationsprozess zu einem Replikat des entsprechenden Shards Ihres Quellclusters, der dieselben Slots besitzt.

Sobald die Änderungen auf der Client-Seite durchgeführt wurden, rufen Sie die `CompleteMigration`-API-Operation auf. Diese API-Operation stuft die ElastiCache-Bereitstellung zur primären Redis-Bereitstellung mit primären und Replikatknoten (sofern relevant) hoch. Nun können Sie die Client-Anwendung veranlassen, Daten ab jetzt nach ElastiCache zu schreiben. Während der Migration können Sie den Status der Replikation prüfen, indem Sie den Befehl [redis-cli INFO](#) für die Redis-Knoten und für den ElastiCache-Primärknoten ausführen.

Schritte zur Migration

Die folgenden Themen beschreiben den Prozess zur Migration der Daten:

- [Vorbereiten Ihrer Redis-Quell- und -Zielknoten für die Migration](#)
- [Testen der Datenmigration](#)
- [Migration starten](#)
- [Verifizieren des Fortschritts der Datenmigration](#)
- [Datenmigration abschließen](#)

Vorbereiten Ihrer Redis-Quell- und -Zielknoten für die Migration

Sie müssen sicherstellen, dass alle vier unten beschriebenen Voraussetzungen erfüllt sind, bevor Sie die Migration mit der ElastiCache-Konsole, der API oder dem AWS-CLI starten.

So bereiten Sie die Redis-Quell- und -Zielknoten für die Migration vor

1. Identifizieren Sie die ElastiCache-Zielbereitstellung und stellen Sie sicher, dass Sie Daten in die Bereitstellung migrieren können.

Eine existierende oder neue ElastiCache-Bereitstellung muss für die Migration die folgenden Voraussetzungen erfüllen:

- Es wird die Redis-Engine-Version ab 5.0.6 verwendet.
- Weder die Verschlüsselung während der Übertragung noch die Verschlüsselung ruhender Daten ist aktiviert.
- Es hat Multi-AZ aktiviert.
- Es ist genug Speicher für die Daten des Redis-Clusters verfügbar. Zum Konfigurieren der richtigen Einstellungen für reservierten Speicher siehe [Verwalten von reserviertem Speicher](#).
- Bei deaktiviertem Cluster-Modus können Sie direkt von Redis ab Version 2.8.21 auf Redis ab Version 5.0.6 migrieren, wenn Sie die CLI verwenden, oder auf Redis ab Version 5.0.6, wenn Sie die CLI oder die Konsole verwenden. Bei aktiviertem Cluster-Modus können Sie direkt von jeder Cluster-modusfähigen Redis-Version auf Redis ab Version 5.0.6 migrieren, wenn Sie die CLI verwenden, oder auf Redis ab Version 5.0.6, wenn Sie die CLI oder die 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 von Open-Source-Redis und der Bereitstellung von ElastiCache für Redis kompatibel sind.

Für die Redis-Replikation müssen mindestens die folgenden Aspekte der ElastiCache-Zielbereitstellung mit der Redis-Konfiguration kompatibel sein:

- Für den Redis-Cluster darf Redis AUTH nicht aktiviert sein.
 - Für den Redis-Konfigurationsparameter `protected-mode` muss `no` eingestellt sein.
 - Wenn die Redis-Konfiguration den Konfigurationsparameter `bind` enthält, sollte sie aktualisiert werden, damit Anforderungen von ElastiCache-Knoten zulässig sind.
 - Die Anzahl der logischen Datenbanken muss für den ElastiCache-Knoten und Ihren Redis-Cluster identisch sein. Dieser Wert wird in der Redis-Konfiguration mit `databases` festgelegt.
 - Redis-Befehle, die Daten ändern, dürfen nicht umbenannt werden, da die Replikation der Daten ansonsten fehlschlägt, z. B. `sync`, `psync`, `info`, `config`, `command` und `cluster`.
 - Sie müssen sicherstellen, dass genug CPU-Kapazität und Speicher verfügbar ist, um die zusätzliche Verarbeitungslast beim Replizieren der Daten vom Redis-Cluster zu ElastiCache zu bewältigen. Diese Last entsteht durch die vom Redis-Cluster erstellte RDB-Datei, die über das Netzwerk zum ElastiCache-Knoten übertragen werden muss.
 - Alle Redis-Instances im Quell-Cluster sollten auf demselben Port ausgeführt werden.
3. Stellen Sie folgendermaßen sicher, dass die Instances eine Verbindung mit ElastiCache herstellen können:
 - Stellen Sie sicher, dass die IP-Adresse jeder Instance privat ist.
 - Weisen Sie die ElastiCache-Bereitstellung zu oder erstellen Sie sie in derselben Virtual Private Cloud (VPC), in der sich Ihr Redis auf Ihrer Instance befindet (empfohlen).
 - Wenn separate VPCs verwendet werden, muss das VPC-Peering so eingerichtet werden, dass Zugriff zwischen den Knoten zulässig ist. Weitere Informationen zum VPC-Peering siehe [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#).
 - Die an die Redis-Instances angefügte Sicherheitsgruppe muss eingehenden Datenverkehr von ElastiCache-Knoten zulassen.
 4. Stellen Sie sicher, dass die Anwendung Datenverkehr nach Abschluss der Migration an ElastiCache-Knoten leiten kann. Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#).

Testen der Datenmigration

Nachdem alle Vorbereitungen durchgeführt wurden, können Sie die Migrationseinrichtung mit der AWS Management Console, der ElastiCache-API oder dem AWS CLI validieren. Das folgende Beispiel illustriert die Verwendung der CLI.

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.

Nachstehend finden Sie ein Beispiel für die Verwendung der CLI.

```
aws elasticache test-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

ElastiCache validiert die Migrationseinrichtung ohne eine tatsächliche Datenmigration.

Migration starten

Nachdem alle Vorbereitungen durchgeführt wurden, können Sie die Datenmigration mit der AWS Management Console, der ElastiCache-API oder dem AWS CLI durchführen. Bei aktiviertem Clustermodus und einer unterschiedlichen Slot-Migration wird vor der Live-Migration ein Resharding durchgeführt. Das folgende Beispiel illustriert die Verwendung der CLI.

Note

Wir empfehlen die Verwendung der `TestMigration`-API zur Validierung der Migrationseinrichtung. Dies ist allerdings völlig optional.

Starten Sie die Migration, indem Sie den Befehl `start-migration` mit den folgenden Parametern aufrufen:

- `--replication-group-id` – Bezeichner der ElastiCache Ziel-Replikationsgruppe
- `--customer-node-endpoint-list` – Eine Liste von Endpunkten mit DNS- oder IP-Adressen und dem Port, an dem Ihr Quell-Redis-Cluster ausgeführt wird. Die Liste kann sowohl bei

deaktiviertem Clustermodus als auch bei aktiviertem Clustermodus nur ein Element enthalten. Wenn die verkettete Replikation aktiviert ist, kann der Endpunkt auf eine Replikat anstelle des Primärknotens im Redis-Cluster zeigen.

Nachstehend finden Sie ein Beispiel für die Verwendung der CLI.

```
aws elasticache start-migration --replication-group-id test-cluster --customer-node-endpoint-list "Address='10.0.0.241',Port=6379"
```

Wenn Sie diesen Befehl ausführen, wird der primäre ElastiCache-Knoten (in jedem Shard) so konfiguriert, dass er ein Replikat Ihrer Redis-Instance wird (im entsprechenden Shard, der dieselben Slots im clusterfähigen Redis besitzt). Der Status des ElastiCache-Clusters wechselt zu migrating (Migration läuft) und die Datenmigration von der Redis-Instance zum ElastiCache-Primärknoten beginnt. In Abhängigkeit vom Umfang der Daten und der Last der Redis-Instance kann die Migration einige Zeit in Anspruch nehmen. Während der Migration können Sie den Fortgang des Prozesses prüfen, indem Sie den Befehl [redis-cli INFO](#) für die Redis-Instance und den ElastiCache-Primärknoten ausführen.

Nach der erfolgreichen Replikation werden alle Schreiboperationen in die Redis-Instances an den ElastiCache-Cluster weitergeleitet. Sie können ElastiCache-Knoten für Lesevorgänge verwenden. Sie können jedoch nicht in den ElastiCache-Cluster schreiben. Wenn andere Replikatknoten mit einem ElastiCache-Primärknoten verbunden sind, können diese Replikatknoten weiterhin vom ElastiCache-Primärknoten replizieren. Auf diese Weise werden alle Daten aus dem Redis-Cluster auf alle Knoten im ElastiCache-Cluster repliziert.

Wenn ein ElastiCache-Primärknoten nicht unmittelbar Replikat der Redis-Instance werden kann, wird dies mehrmals versucht, bevor die Rückstufung zum Primär erfolgt. Der Status des ElastiCache-Clusters wechselt dann zu available (Verfügbar) und ein Replikationsgruppenereignis zum Fehlschlagen des Initiierens der Migration wird gesendet. 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 Redis-Quell- und -Zielknoten für die Migration](#) befolgt wurden.
- Stellen Sie sicher, dass die Routing-Konfiguration für die VPC und die Subnetze Datenverkehr zwischen ElastiCache-Knoten und den Redis-Instances erlaubt.

- Stellen Sie sicher, dass die an die Redis-Instances angefügte Sicherheitsgruppe eingehenden Datenverkehr von ElastiCache-Knoten zulässt.
- Prüfen Sie die Redis-Protokolle für die Redis-Instances auf weitere Informationen zu replikationsspezifischen Fehlern.

Verifizieren des Fortschritts der Datenmigration

Nach Beginn der Datenmigration können Sie den Verlauf folgendermaßen überwachen:

- Verifizieren Sie, dass der `master_link_status` von Redis im Befehl `INFO` für den/die ElastiCache-Primärknoten up lautet. Sie finden diese Informationen auch in der ElastiCache-Konsole. Wählen Sie den Cluster aus und beachten Sie unter CloudWatch metrics (CloudWatch-Metriken) den Primary Link Health Status (Zustand des Masterlinks). Die Daten sind synchronisiert, sobald der Wert 1 erreicht hat.
- Sie können ermitteln, ob das ElastiCache-Replikat den Status online hat, indem Sie den Befehl `INFO` auf den Redis-Instances ausführen. Dadurch werden auch Informationen zur Replikationsverzögerung bereitgestellt.
- Verifizieren Sie niedrige Client-Ausgabepufferverfügbarkeit, indem Sie den Redis-Befehl [CLIENT LIST](#) auf den Redis-Instances ausführen.

Nach Abschluss der Datenmigration sind die Daten mit allen neuen Schreibvorgängen auf dem/den Primärknoten Ihres Redis-Clusters synchronisiert.

Datenmigration abschließen

Wenn Sie bereit sind, zum ElastiCache-Cluster umzuschalten, verwenden Sie den CLI-Befehl `complete-migration` 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
```

Wenn Sie diesen Befehl ausführen, stoppt der ElastiCache-Primärknoten (in jedem Shard) die Replikation von der Redis-Instance und übergibt sie an den primären Knoten. 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`. Sie können Ihre Anwendung jetzt zu ElastiCache-Schreib- und Leseoperationen leiten. Der ElastiCache Clusterstatus sollte sich von Migration auf Verfügbar ändern.

Wenn die Umstufung zum Master fehlschlägt, repliziert der ElastiCache-Primärknoten weiterhin von der Redis-Instance. Der ElastiCache-Cluster behält den Status `migrating` (Migration läuft) und eine Replikationsgruppenereignismeldung zum Fehler wird 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. In diesem Fall müssen Sie sicherstellen, dass der primäre ElastiCache-Knoten aus der Redis-Instance replizieren kann und die beiden Komponenten synchronisiert 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 ersetzt wird. Sie können den Schritt zum Abschließen der Migration wiederholen, nachdem der Austausch abgeschlossen wurde.


Durchführen einer Online-Datenmigration mithilfe der Konsole

Sie können die AWS Management Console verwenden, um die Daten aus dem Cluster zum 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 Redis-Cluster oder wählen Sie einen vorhandenen Cluster aus. Stellen Sie sicher, dass der Cluster die folgenden Anforderungen erfüllt:
 - Die Redis-Engine-Version muss mindestens 5.0.6 oder höher sein.
 - Für den Redis-Cluster darf Redis AUTH nicht aktiviert sein.

- Für den Redis-Konfigurationsparameter `protected-mode` muss `no` eingestellt sein.
 - Wenn die Redis-Konfiguration den Konfigurationsparameter `bind` enthält, sollte sie aktualisiert werden, damit Anforderungen von ElastiCache-Knoten zulässig sind.
 - Die Anzahl der Datenbanken sollte für den ElastiCache-Knoten und Ihren Redis-Cluster identisch sein. Dieser Wert wird in der Redis-Konfiguration mit `databases` festgelegt.
 - Redis-Befehle, die Daten ändern, dürfen nicht umbenannt werden, da die Replikation der Daten ansonsten fehlschlägt.
 - Sie müssen sicherstellen, dass genug CPU-Kapazität und Speicher verfügbar ist, um die zusätzliche Verarbeitungslast beim Replizieren der Daten vom Redis-Cluster zu ElastiCache zu bewältigen. Diese Last entsteht durch die vom Redis-Cluster erstellte RDB-Datei, die über das Netzwerk zum ElastiCache-Knoten übertragen werden muss.
 - Der Cluster hat den Status `available` (Verfügbar).
3. Wählen Sie bei ausgewähltem Cluster Migrieren von Daten von Endpoint für Aktionen aus.
 4. Geben Sie in das Dialogfeld Daten vom Endpunkt migrieren die IP-Adresse und den Port ein, an dem der 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 Availability Zones

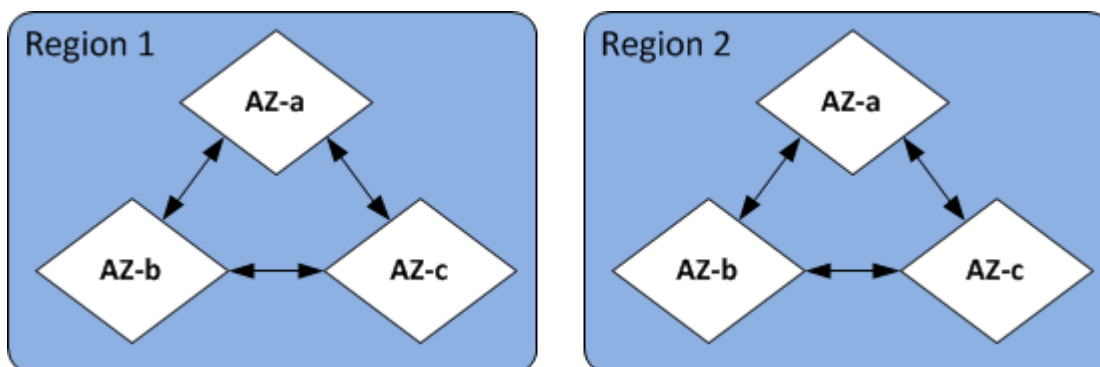
AWS Cloud-Computing-Ressourcen befinden sich in hochverfügbaren Rechenzentrumseinrichtungen. Die Einrichtungen dieser Rechenzentren befinden sich an verschiedenen Standorten, um die Skalierbarkeit und Zuverlässigkeit zu erhöhen. Diese Standorte sind nach Regionen und Availability Zones kategorisiert.

AWS Regionen sind groß und weit verbreitert in separate geografische Standorte. Availability Zones sind unterschiedliche Standorte innerhalb einer - AWS Region, die vor Ausfällen in anderen Availability Zones geschützt sind. Sie bieten kostengünstige Netzwerkkonnektivität mit geringer 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

- [Lokalisieren Ihrer Knoten](#)
- [Unterstützte Regionen und Endpunkte](#)
- [Verwenden von Local Zones mit ElastiCache](#)
- [Verwenden von Outposts](#)

Lokalisieren Ihrer Knoten

Amazon ElastiCache unterstützt das Auffinden aller Knoten eines Clusters in einer oder mehreren Availability Zones (AZs). Wenn Sie Ihre Knoten in mehreren AZs platzieren möchten (empfohlen), ElastiCache ermöglicht Ihnen außerdem, entweder die AZ für jeden Knoten auszuwählen oder die Auswahl für Sie ElastiCache zu erlauben.

Durch Ansiedeln von Clustern oder Knoten in verschiedenen AZs, eliminieren Sie das Risiko, dass Ihr gesamtes System aufgrund eines Fehlers, wie z. B. eines Stromausfalls, in einer AZ ausfällt. Tests haben gezeigt, dass der Latenzunterschied zwischen der Ansiedlung aller Knoten in einer AZ und ihrer Verteilung über mehrere AZ hinweg nicht signifikant ist.

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:

- [Erstellen eines Clusters](#)
- [Einen ElastiCache Cluster ändern](#)
- [Knoten zu einem Cluster hinzufügen](#)

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 starten, 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 Serverless-Caches replizieren Daten automatisch über mehrere Availability Zones hinweg (außer us-west-1, wo Daten in zwei Availability Zones repliziert werden), um eine hohe Verfügbarkeit zu gewährleisten. Beim Entwerfen eines eigenen ElastiCache Clusters können Sie Ihre Knoten in verschiedenen AZs starten, um eine Fehlertoleranz zu erreichen. Weitere Informationen zu Regionen und Availability Zones erhalten Sie unter [Auswahl von Regionen und Availability Zones](#) oben in diesem Thema.

Regionen, in denen unterstützt ElastiCache 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 (VAE)“ 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 (USA-West) us-gov-west-1 | elasticache.us- gov-west-1.a mazonaws.com | HTTPS |
| AWS GovCloud (USA-Ost) us-gov-east-1 | elasticache.us- gov-east-1.a mazonaws.com | HTTPS |

Informationen zur Verwendung der AWS GovCloud (USA) mit ElastiCache finden Sie unter [Services in der AWS GovCloud \(USA\)-Region: ElastiCache](#)

Einige Regionen unterstützen eine Untermenge von Knotentypen. Eine Tabelle der unterstützten Knotentypen nach AWS Region finden Sie unter [Unterstützte Knotentypen nach AWS-Region](#).

Eine Tabelle der AWS Produkte und Services nach Regionen finden Sie unter [Produkte und Services nach Regionen](#).

Verwenden von Local Zones mit ElastiCache

Eine Local Zone ist eine Erweiterung einer AWS-Region, die geografisch in der Nähe Ihrer Benutzer liegt. Sie können jede Virtual Private Cloud (VPC) aus der übergeordneten AWS-Region in Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es der Local Zone zuweisen. Wenn Sie ein Subnetz in einer lokalen Zone erstellen, wird Ihre VPC auf diese lokale Zone erweitert. Das Subnetz in der lokalen Zone funktioniert genauso wie andere Subnetze in Ihrer VPC.

Durch die Verwendung von Local Zones können Sie Ressourcen wie beispielsweise einen ElastiCache-Cluster an mehreren Standorten in der Nähe Ihrer Benutzer platzieren.

Wenn Sie einen ElastiCache-Cluster erstellen, können Sie ein Subnetz in einer Local 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](#).

Die Lokale Zone wird durch einen Regionscode AWS dargestellt, gefolgt von einer ID, die den Standort angibt, z. B. `us-west-2-lax-1a`.

Zu diesem Zeitpunkt sind die verfügbaren Local Zones `us-west-2-lax-1a` und `us-west-2-lax-1b`.

Die folgenden Einschränkungen gelten für 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 Local Zone in der Amazon-EC2-Konsole.

Weitere Informationen finden Sie unter [Aktivieren von Local Zones](#) im Benutzerhandbuch zu Amazon EC2.

2. Erstellen Sie ein Subnetz in der Local Zone.

Weitere Informationen finden Sie unter [Erstellen eines Subnetzes in Ihrer VPC](#) im Amazon-VPC-Benutzerhandbuch.

3. Erstellen Sie eine ElastiCache-Subnetzgruppe in der Local Zone.

Wenn Sie eine ElastiCache-Subnetzgruppe erstellen, wählen Sie die Gruppe der Availability Zones für die Local Zone aus.

Weitere Informationen finden Sie unter [Erstellen einer Subnetzgruppe](#) im Benutzerhandbuch zu ElastiCache.

- Erstellen Sie einen ElastiCache-for-Redis-Cluster, der das ElastiCache-Subnetz in der Local Zone verwendet. Weitere Informationen finden Sie in den folgenden Themen:
 - [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
 - [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Verwenden von Outposts

AWS Outposts ist ein vollständig verwalteter Service, der AWS Infrastruktur, Dienste, APIs und Tools auf Kundenstandorte ausdehnt. Durch den lokalen Zugriff auf die AWS verwaltete Infrastruktur ermöglicht AWS Outposts seinen 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.

Verwenden von Outposts mit der Redis-Konsole

- Melden Sie AWS Management Console sich bei ElastiCache [der](#) an und öffnen Sie die Konsole unter <https://console.aws.amazon.com/elasticache/>.

2. Wählen Sie im Navigationsbereich Redis-Caches aus.
3. Wählen Sie Redis-Cache erstellen aus.
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 VPC-ID 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. Unter Redis-Einstellungen:
 - a. Name: Geben Sie einen Namen für den Redis-Cluster ein
 - b. Beschreibung: Geben Sie eine Beschreibung für den Redis-Cluster ein.
 - c. Kompatibilität der Engine-Version: Die Engine-Version basiert auf der AWS Outpost-Region
 - d. Port: Akzeptieren Sie den Standardport 6379. Wenn Sie aus irgendeinem Grund einen anderen Port verwenden möchten, geben Sie die betreffende 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. Der Portierungsassistent für .NET-für-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 Replikat: 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
 - VPC-ID: Die VPC-ID sollte mit der Outpost-VPC übereinstimmen. Wenn Sie eine VPC auswählen, die keine Subnetz-IDs auf den Outposts enthält, wird eine leere Liste 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 keine Subnetz-IDs verfügbar sind, müssen Sie welche erstellen. Weitere Informationen finden Sie unter [Ein Subnetz erstellen](#).
- b. Wählen Sie Erstellen aus.

Anzeige der Details des Outpost-Clusters

Wählen Sie auf der Redis-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 Outpost dar, wobei ein ARN (Amazon Resource Name) und die AWS Ressourcennummer verwendet werden.
- Name des Außenpostens: Der Name des Außenpostens AWS .

Outposts mit der AWS CLI verwenden

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 die 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.

So laden Sie den CLI herunter, installieren und konfigurieren ihn

1. Laden Sie die AWS CLI von der [AWS Command Line Interface-Webseite](#) herunter.
2. Folgen Sie den Anweisungen zur [Installation der AWS CLI](#) und [zur Konfiguration der AWS CLI](#) im AWS Command Line Interface Benutzerhandbuch.

Verwenden der AWS CLI mit Outposts

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 Details zur Verwaltung der verschiedenen Komponenten Ihrer ElastiCache Implementierung.

Themen

- [Snapshot und Wiederherstellung](#)
- [Engine-Versionen und Upgrades](#)
- [ElastiCache Best Practices und Caching-Strategien](#)
- [Verwalten Ihrer selbst entworfenen Cluster](#)
- [Skalierung ElastiCache für Redis](#)
- [Erste Schritte mit JSON in ElastiCache for Redis](#)
- [Markieren Ihrer ElastiCache-Ressourcen](#)
- [Verwendung von Amazon ElastiCache Well-Architected Lens](#)
- [Häufige Schritte und bewährte Methoden zur Fehlerbehebung](#)
- [Zusätzliche Schritte zur Fehlerbehebung](#)

Snapshot und Wiederherstellung

ElastiCache Amazon-Caches, auf denen Redis ausgeführt wird, 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 Redis erstellen und ihn mit Daten aus einem Backup füllen. Mit ElastiCache können Sie Backups mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) und der API verwalten. ElastiCache

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](#)
- [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 Redis oder Serverless Memcached ausgeführt werden.
- Bei Redis-Clustern (Cluster-Modus deaktiviert) werden Sicherung und Wiederherstellung auf `cache.t1.micro`-Knoten nicht unterstützt. Alle anderen Cache-Knotentypen werden unterstützt.
- Bei Redis-Clustern (Cluster-Modus 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 von Redis selbst entworfene Cluster können Sie nicht mehr als 20 manuelle Backups pro Knoten im Cluster erstellen.
- Redis (Cluster-Modus aktiviert) unterstützt nur das Erstellen von Sicherungen auf Cluster-Ebene (für die API oder CLI, die Replikationsgruppenebene). Redis (Cluster-Modus aktiviert) unterstützt nur das Erstellen von Sicherungen auf Shard-Ebene (für die API oder CLI auf der Knotengruppenebene).

- Während des Backup-Vorgangs können Sie keine anderen API- oder CLI-Operationen im serverlosen Cache ausführen. Sie können während des Backups API- oder CLI-Operationen auf einem selbst entworfenen Cluster ausführen.
- Wenn Sie Caches mit Daten-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. Selbst entworfene Cluster sind nicht mit ElastiCache und Memcached verfügbar, sondern mit und Redis. ElastiCache

Nachfolgend sind Richtlinien zur Verbesserung der Backup-Leistung für selbst entworfene Cluster aufgeführt.

- Legen Sie den Parameter `reserved-memory-percent` fest – Zur Einschränkung übermäßiger Auslagerung wird empfohlen, den Parameter `reserved-memory-percent` festzulegen. Dieser Parameter hindert Redis am Verbrauch des gesamten verfügbaren Arbeitsspeichers des Knotens und kann zur Verringerung der Auslagerung beitragen. Die Leistung lässt sich möglicherweise einfach durch Verwenden eines größeren Knotens verbessern. Weitere Informationen zu den Parametern `reserved-memory` und zu `reserced-memory-persent` finden Sie unter [Verwalten von reserviertem Speicher](#).
- Erstellen von Sicherungen einer Read Replica Wenn Sie Redis in einer Knotengruppe mit mehreren Knoten ausführen, können Sie eine Sicherung des primären Knotens oder einer der Read Replicas erstellen. Aufgrund der während BGSAVE erforderlichen Systemressourcen wird empfohlen, Sicherungen von einer der Read Replicas zu erstellen. Während die Sicherung von der Replikation erstellt wird, wirken sich die BGSAVE-Ressourcenanforderungen nicht auf den primären Knoten aus. 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 vor dem Löschen der Replikationsgruppe die neuesten Redis-Daten erfasst werden.

Planen automatischer Backups

Sie können automatische Backups für jeden serverlosen 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](#).

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.

Sie können automatische Backups aktivieren oder deaktivieren, wenn Sie entweder einen neuen Cache erstellen oder einen vorhandenen Redis-Cache aktualisieren, indem Sie die ElastiCache Konsole AWS CLI, die oder die ElastiCache API verwenden. Dazu aktivieren Sie das Kontrollkästchen Automatische Backups aktivieren im Abschnitt Erweiterte Redis-Einstellungen.

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 ein manuelles Backup eines Caches mithilfe der AWS Management Console AWS CLI, der oder der ElastiCache API erstellen.

Erstellen einer manuellen Sicherung (Konsole)

So erstellen Sie ein manuelles Backup eines Caches (Konsole)

1. Melden Sie sich bei der Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Redis-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 von Ihnen gesicherte Cluster über keine Replikatknoten verfügt, lautet der Name des Clusters, den Sie sichern `--cache-cluster-id` (z. B. *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 kann beispielsweise *mycluster-002* sein.

Verwenden Sie diesen Parameter nur zur Sicherung eines Redis-Clusters (Cluster-Modus deaktiviert).

- `--replication-group-id` – Name des Redis-Clusters (Cluster-Modus aktiviert) (CLI/API: eine Replikationsgruppe), der als Quelle für die Sicherung verwendet werden soll. Verwenden Sie diesen Parameter zur Sicherung eines Redis-Clusters (Cluster-Modus aktiviert).
- `--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: Sichern eines Redis-Clusters (Clustermodus deaktiviert) ohne Replikatknoten

Mit dem folgenden AWS CLI Vorgang wird das Backup `bkup-20150515` aus dem Redis-Cluster (Clustermodus deaktiviert) erstellt `myNonClusteredRedis`, der keine Lesereplikate 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: Sichern eines Redis-Clusters (Clustermodus deaktiviert) mit Replikatknoten

Mit dem folgenden AWS CLI Vorgang wird das Backup *bkup-20150515* aus dem Redis-Cluster (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: Sichern eines Redis-Clusters (Cluster-Modus 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: Sichern eines Clusters für Redis (Clustermodus aktiviert)

Der folgende AWS CLI Vorgang erstellt das Backup `bkup-20150515` aus dem Redis-Cluster (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: Sichern eines Redis-Clusters (Cluster-Modus 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 mit der ElastiCache Konsole, der oder der AWS CLI ElastiCache API ein endgültiges Backup erstellen.

Erstellen einer endgültigen Sicherung (Konsole)

Sie können ein endgültiges Backup erstellen, wenn Sie einen serverlosen Redis-Cache oder einen selbst entworfenen Cluster mithilfe der Konsole löschen. ElastiCache

Um beim Löschen eines Caches eine endgültige Sicherung zu erstellen, wählen Sie im Löschdialogfeld unter Backup erstellen die Option Ja aus und geben Sie der Sicherung einen Namen.

Verwandte Themen

- [Verwendung von 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 serverlosen Caches](#)
- [Beim Löschen eines von Redis selbst entworfenen Clusters ohne Read Replicas](#)
- [Bei Löschung eines Redis-Clusters mit Read Replicas](#)

Beim Löschen eines serverlosen Caches

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 unter [delete-serverless-cache](#) in der AWS CLI -Befehlsreferenz.

Beim Löschen eines von Redis selbst entworfenen 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 unter [delete-cache-cluster](#) in der AWS CLI -Befehlsreferenz.

Bei Löschung eines 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 unter [describe-replication-groups](#) 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 mit dem 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 die CLI-Operation `describe-serverless-cache-snapshots`, um eine Liste von Serverless-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 unter [describe-serverless-cache-snapshots](#) in der Befehlsreferenz.
AWS CLI

Beschreiben von selbst entworfenen Cluster-Backups (AWS CLI)

Verwenden Sie die CLI-Operation `describe-snapshots`, um eine Liste von Backups selbst entworfener Cluster 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. Anleitungen zum Exportieren Ihres Backups 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 unter <https://console.aws.amazon.com/elasticache/> bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole.
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 entworfenes Cluster-Backups (AWS CLI)

Verwenden Sie die Operation `copy-snapshot`, um ein Backup eines selbst entworfenes 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. Nur für serverlose Redis-Caches und selbst entworfene Redis-Cluster verfügbar. 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 Backups ElastiCache für Redis in einen Amazon Simple Storage Service (Amazon S3) -Bucket, sodass Sie von außerhalb ElastiCache darauf zugreifen können. Sie können ein Backup mithilfe der ElastiCache Konsole, der AWS CLI, oder der 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 [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.

Wichtig

- 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](#).
- Das Exportieren eines Backups ist für selbst entworfene Redis-Cluster, Serverless Redis und 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.

Schritt 1: Einen Amazon-S3-Bucket erstellen

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 Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie 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-Buckets muss DNS-konform sein. Andernfalls ElastiCache kann nicht auf Ihre Backup-Datei zugreifen. Die Regeln für die DNS-Konformität lauten:

- 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 Erstellen aus.

Weitere Informationen zum Erstellen eines Amazon-S3-Buckets finden Sie unter [Erstellen von Buckets](#) im Handbuch für Amazon Simple Storage Service.

Schritt 2: ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket gewähren

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. Daher wird sehr dazu geraten, IAM-Richtlinien einzurichten, um unbefugten Zugriff auf diesen

Amazon-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 Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie 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 [Schritt 1: Einen Amazon-S3-Bucket erstellen](#) erstellte S3-Bucket sein.
3. Wählen Sie die Registerkarte Permissions (Berechtigungen) und unter Permissions (Berechtigungen) Access control list (ACL) (Zugriffssteuerungsliste (ACL)) und dann Edit (Bearbeiten) aus.
4. Fügen Sie die Fügen Sie die Kanonische Empfänger-ID
540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353 mit den folgenden Optionen hinzu:
 - Objekte – Auflisten oder Schreiben
 - Bucket ACL: Lesen oder Schreiben

 Note

- Für die GovCloud PDT-Region lautet die Canonical ID.
40fa568277ad703bd160f66ae4f83fc9dfdfd06c2f1b5060ca22442ac3ef8be6
- Für die GovCloud OSU-Region lautet die kanonische ID.
c54286759d2a83da9c480405349819c993557275cf37d820d514b42da6893f5c

5. Wählen Sie Speichern.

Schritt 3: Exportieren Sie ein Backup ElastiCache

Jetzt haben Sie Ihren S3-Bucket erstellt und ElastiCache Zugriffsberechtigungen erteilt. Als Nächstes können Sie die ElastiCache Konsole, die AWS CLI oder die ElastiCache API verwenden, um Ihren Snapshot dorthin zu exportieren. In den folgenden, zusätzlichen S3-spezifischen IAM-Berechtigungen

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 UTF-8-codierbar sein.

ElastiCache fügt eine Instanz-ID und `.rdb` zu dem Wert hinzu, den Sie hier eingeben. 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 [Schritt 1: Einen Amazon-S3-Bucket erstellen](#) 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 [Schritt 2: ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#).

- 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 [Schritt 2: ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#) zurück, um die angegebenen Berechtigungen hinzuzufügen, und exportieren Sie Ihre Sicherung erneut.

- ElastiCache hat keine LESE-Rechte %s für den S3-Bucket erhalten.

Lösung: Fügen Sie Leseberechtigungen für den Bucket hinzu.

- ElastiCache hat keine SCHREIBBERECHTIGUNG %s für den S3-Bucket erhalten.

Lösung: Fügen Sie Schreibberechtigungen für den Bucket hinzu.

- ElastiCache hat keine READ_ACP-Berechtigungen %s für den S3-Bucket erhalten.

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 die Sicherung mit der CLI-Operation `export-serverless-cache-snapshot` mit den folgenden Parametern zu einem Amazon-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 die Sicherung mit der CLI-Operation `copy-snapshot` mit den folgenden Parametern zu einem Amazon-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 UTF-8-codierbar sein.

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 [Schritt 2: ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket gewähren](#).

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 in einem neuen serverlosen Cache oder einem selbst entworfenen Cluster wiederherstellen.

Wiederherstellen eines Backups in einem Serverless-Cache (Konsole)

Note

ElastiCache Serverless unterstützt RDB-Dateien, die mit Redis-Versionen zwischen 5.0 und der neuesten verfügbaren Version kompatibel sind.

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/.](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 <https://console.aws.amazon.com/elasticache/> an AWS Management Console und öffnen Sie die ElastiCache Konsole.
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 Redis-Versionen zwischen 5.0 und der neuesten verfügbaren Version kompatibel sind.

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 einer Sicherung verwendet `create-serverless-cache` und aus dieser 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
```

Für Windows:

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 serverloses Redis-Cache-Backup wiederherstellen, und Sie können auch einen von Redis selbst entworfenen Cluster wiederherstellen.

Sie können ein serverloses Redis-Cache-Backup auf zwei Arten wiederherstellen.

- Mithilfe dieses Vorgangs können Sie eine Wiederherstellung auf einem Redis-Cluster mit einem Knoten (Clustermodus deaktiviert) durchführen. AWS CLI `create-cache-cluster`
- Sie können auf einen Redis-Cluster mit Read Replicas (eine Replikationsgruppe) wiederherstellen. Zu diesem Zweck können Sie entweder Redis (Clustermodus deaktiviert) oder Redis (Clustermodus aktiviert) für den Vorgang verwenden. AWS CLI `create-replication-group` In diesem Fall führen Sie das Seeding der Wiederherstellung mit einer Redis-`.rdb`-Datei durch. Weitere Informationen zum Seeding eines neuen selbst entworfenen Clusters finden Sie unter [Seeding eines neuen selbst entworfenen Clusters mit einem extern erstellten Backup](#).

Sie können eine Redis-Sicherung (Cluster-Modus deaktiviert) auf zwei Arten wiederherstellen.

- Sie können mithilfe des Vorgangs eine Wiederherstellung auf einem Redis-Cluster mit einem Knoten (Clustermodus deaktiviert) durchführen. AWS CLI `create-cache-cluster`
- Sie können auf einen Redis-Cluster mit Read Replicas (eine Replikationsgruppe) wiederherstellen. Zu diesem Zweck können Sie entweder Redis (Clustermodus deaktiviert) oder Redis (Clustermodus aktiviert) für den Vorgang verwenden. AWS CLI `create-replication-group` In diesem Fall führen Sie das Seeding der Wiederherstellung mit einer Redis-`.rdb`-Datei durch. Weitere Informationen zum Seeding eines neuen selbst entworfenen Clusters finden Sie unter [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 mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API löschen.

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 aus.
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 unter [delete-serverless-cache-snapshot](#) in der AWS CLI - Befehlsreferenz.

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 der AWS 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 AWS CLI, der oder der ElastiCache API können Sie Kostenzuordnungs-Tags zu Ihren Backups hinzufügen, auflisten, ändern, entfernen oder kopieren. Weitere Informationen finden Sie unter [Überwachung von Kosten mit Kostenzuordnungs-Tags](#).

Seeding eines neuen selbst entworfenen Clusters mit einem extern erstellten Backup

Wenn Sie einen neuen selbst entworfenen Redis-Cluster erstellen, können Sie ihn mit Daten aus einer RDB-Backup-Datei von Redis starten. Das Seeding des Clusters ist nützlich, wenn Sie derzeit eine Redis-Instanz außerhalb von verwalteten ElastiCache und Ihren neuen, ElastiCache für Redis selbst entworfenen Cluster mit Ihren vorhandenen Redis-Daten füllen möchten.

Informationen zum Seeding eines neuen, selbst entworfenen Redis-Clusters aus einem in Amazon ElastiCache erstellten Redis-Backup finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#)

Wenn Sie einen neuen selbst entworfenen Redis-Cluster anhand einer RDB-Datei von Redis starten, können Sie folgende Aufgaben ausführen:

- Führen Sie ein Upgrade von einem nicht partitionierten Cluster auf einen selbst entworfenen Redis-Cluster (Cluster-Modus aktiviert) mit Redis Version 3.2.4 durch.
- Geben Sie eine Anzahl der Shards (in der API und CLI Knotengruppen genannt) im neuen selbst entworfenen Cluster an. 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 zu einem kleineren Knotentyps skalieren, stellen Sie sicher, dass der Speicher des neuen Knotentyps für Ihre Daten und den Redis-Overhead ausreicht. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#).
- Verteilen Sie Ihre Schlüssel in den Slots des neuen Redis-Clusters (Cluster-Modus aktiviert) anders als in dem Cluster, der zur Erstellung der Sicherungsdatei verwendet wurde.

Note

Sie können einen Redis-Cluster (Cluster-Modus deaktiviert) nicht von einer .rdb-Datei aus starten, die von einem Redis-Cluster (Cluster-Modus aktiviert).

Important

- Sie müssen sicherstellen, dass die Redis-Sicherungsdaten nicht die Ressourcen des Knotens überschreiten. Sie können z. B. keine .rdb-Datei mit 5 GB Redis-Daten zu einem 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 [Spezifische Parameter für den Redis-Knotentyp](#) und zu [ElastiCache Amazon-Produktfunktionen und -details](#).

- Sie können eine .rdb-Redis-Datei nur mit serverseitiger Amazon-S3-Verschlüsselung (SSE-S3) verschlüsseln. Weitere Informationen finden Sie unter [Schützen von Daten mithilfe serverseitiger Verschlüsselung](#).

Im Folgenden finden Sie Themen, die Sie durch die Migration Ihres Redis-Clusters von extern für Redis zu ElastiCache ElastiCache für Redis führen.

Migration zu for Redis ElastiCache

- [Schritt 1: Erstellen einer Redis-Sicherung](#)
- [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 einer Redis-Sicherung

Um das Redis-Backup zu erstellen, um Ihre ElastiCache for Redis-Instanz zu starten

1. Stellen Sie eine Verbindung zu Ihrer vorhandenen Redis-Instance her.
2. Führen Sie entweder die Redis-Operation BGSAVE oder SAVE zum Erstellen einer Sicherung aus. Achten Sie auf den Speicherort der .rdb-Datei.

BGSAVE ist asynchron und blockiert während der Verarbeitung keine anderen Clients. Weitere Informationen finden Sie unter [BGSAVE](#) auf der Redis-Website.

SAVE ist synchron und blockiert andere Vorgänge, bis sie Verarbeitung abgeschlossen ist. Weitere Informationen finden Sie unter [SAVE](#) auf der Redis-Website.

Zusätzliche Informationen zum Erstellen einer Sicherung finden Sie unter [Redis Persistence](#) auf der Redis-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 Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie 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-Buckets muss DNS-konform sein. Andernfalls ElastiCache kann nicht auf Ihre Backup-Datei zugreifen. Die Regeln für die DNS-Konformität lauten:

- 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 for Redis-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 erlauben, dass der Bucket und dessen Inhalt verwendet werden können, um Ihren neuen Redis-Cluster zu starten.

So fügen Sie einem Amazon S3 Bucket einen Ordner hinzu

1. Melden Sie sich bei der Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie 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 Speichern.

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 einer Redis-Sicherung](#) 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 Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie 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 Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie 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 eine der kanonischen IDs in der folgenden Liste nicht als Benutzer aufgelistet wird, führen Sie die folgenden Schritte aus:
 - a. Wählen Sie unter Zugriff für andere AWS Konten die Option Zuschussempfänger 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, AWS GovCloud (US) damit Sie es auf einen Redis-Cluster herunterladen können. 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)
 - Read/write-object-ACL-permissions (Lese-/Schreibberechtigungen für Objekt-ACL)
- d. Wählen Sie Speichern.
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 Amazon S3 S3-Konsole an AWS Management Console und öffnen Sie sie 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 für die 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 aus.

Schritt 5: 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 [Erstellen eines Clusters](#) oder [Erstellen einer Redis-Replikationsgruppe von Grund auf](#). Achten Sie darauf, als Cluster-Engine Redis auszuwählen.

Die Methode, mit der Sie angeben, ElastiCache wo sich das Redis-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 For-Redis-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 Seed-RDB-Datei-S3-Speicherort den Amazon-S3-Pfad für die Datei(en) 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.

- Mit dem AWS CLI

Verwenden Sie bei Einsatz der Operation `create-cache-cluster` oder `create-replication-group` den Parameter `--snapshot-arns`, um einen vollqualifizierten ARN für jede .rdb-Datei anzugeben. z. B. `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. Der ARN muss auf die in Amazon S3 gespeicherten Sicherungsdateien aufgelöst werden.

- Mithilfe der ElastiCache API

Wenn Sie die `CreateReplicationGroup` ElastiCache API-Operation `CreateCacheCluster` oder verwenden, verwenden Sie den Parameter `SnapshotArns` um einen vollqualifizierten ARN für jede .rdb-Datei anzugeben. z. B. `arn:aws:s3:::myBucket/myFolder/myBackupFilename`.rdb. Der ARN muss auf die in Amazon S3 gespeicherten Sicherungsdateien aufgelöst werden.

Important

Beim Seeding eines Redis-Clusters (Cluster-Modus aktiviert) müssen Sie jede Knotengruppe (Shard) in dem neuen Cluster oder der Replikationsgruppe konfigurieren. Dazu verwenden Sie den Parameter `--node-group-configuration` (API: `NodeGroupConfiguration`). Weitere Informationen finden Sie hier:

- CLI: [create-replication-group](#) in der Referenz AWS CLI
- API: [CreateReplicationGruppe](#) in der API-Referenz ElastiCache

Während des Vorgangs zur Cluster-Erstellung werden die Daten in der Redis-Sicherung 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 oder ElastiCache API verwenden, um Ereignismeldungen abzurufen. Weitere Informationen finden Sie unter [Anzeigen von ElastiCache-Ereignissen](#).

Engine-Versionen und Upgrades

Dieser Abschnitt beschreibt die unterstützten Redis-Engine-Versionen, und wie ein Upgrade dafür ausgeführt wird.

Themen

- [Engine-Versionen und Upgrades](#)
- [Unterstützte ElastiCache-for-Redis-Versionen](#)
- [Zeitplan für das Lebensende der Redis-Versionen](#)
- [So führen Sie ein Upgrade von Engine-Versionen aus](#)
- [Beheben blockierter Redis-Engine-Upgrades](#)
- [Wesentliche Versionsverhalten und Kompatibilitätsunterschiede](#)

Engine-Versionen und Upgrades

Versionen von ElastiCache für Redis werden mit einer semantischen Version identifiziert, die eine MAJOR- und eine MINOR-Komponente umfasst. In Redis 6.2 ist beispielsweise die Hauptversion 6 und die Unterversion 2. Beim Betrieb selbst entworfener Cluster macht ElastiCache für Redis auch die PATCH-Komponente verfügbar, z. B. Redis 6.2.1. Die Patch-Version ist dann 1.

MAJOR-Versionen stehen für API-inkompatible Änderungen und MINOR-Versionen für neue Funktionen, die abwärtskompatibel hinzugefügt wurden. PATCH-Versionen stehen für abwärtskompatible Fehlerbehebungen und nicht funktionale Änderungen.

Versionsverwaltung für ElastiCache Serverless

ElastiCache Serverless wendet automatisch die neueste MINOR- und PATCH-Softwareversion auf Ihren Cache an, ohne dass dies Auswirkungen oder Ausfallzeiten für 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 Ihren Cache auf die neueste Hauptversion aktualisieren, indem Sie ihn mithilfe der Konsole, CLI oder API ändern und die neueste Engine-Version auswählen.

Versionsverwaltung für selbst entworfene ElastiCache-Cluster

Wenn Sie mit selbst entworfenen ElastiCache-Clustern arbeiten, können Sie steuern, wann die Software Ihres Cache-Clusters auf neue Versionen aktualisiert wird, die von ElastiCache unterstützt werden. Sie können steuern, wann Ihr Cache auf die neuesten verfügbaren MAJOR-, MINOR- und PATCH-Versionen aktualisiert werden soll. Sie starten Engine-Versions-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 Ihres Cache-Clusters auf neue Versionen aktualisiert wird, die von ElastiCache unterstützt werden. 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.

Sie starten Engine-Version-Upgrades für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie ihn bzw. sie ändern und eine neue Engine-Version angeben. Weitere Informationen finden Sie hier:

- [Ändern von Clustern](#)
- [Ändern einer Replikationsgruppe](#)

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 ElastiCache Serverless.

Beachten Sie bei Upgrades eines selbst entworfenen Clusters die folgenden Ausführungen.

- Versionsverwaltung der Engine ist so entwickelt, dass Sie so viel Kontrolle wie möglich darüber haben, wie Patchen erfolgt. ElastiCache behält sich jedoch das Recht vor, Ihren Cluster in Ihrem Namen für den unwahrscheinlichen Fall einer kritischen Sicherheitslücke im System oder der Cache-Software zu patchen.
- Ab Redis 6.0 bietet ElastiCache für Redis eine einzelne Version für jede Redis-OSS-Nebenversion anstelle mehrerer Patch-Versionen an.
- Ab Redis Engine Version 5.0.6 können Sie Ihre Cluster-Version mit minimalen Ausfallzeiten 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 identisch, kann jedoch längere Failover-Zeit während der DNS-Ausbreitung (30 Sek. - 1 Min.) verursachen.
- Ab Redis 7 unterstützt ElastiCache for Redis nicht das Umschalten zwischen Redis (Cluster-Modus deaktiviert) und Redis (Cluster-Modus aktiviert).
- Der Upgradeprozess der Amazon-ElastiCache-for-Redis-Engine ist darauf ausgelegt, Ihre vorhandenen Daten bestmöglich zu bewahren und erfordert eine erfolgreiche Redis-Replikation.
- Beim Upgrade der Engine beendet ElastiCache for Redis bestehende Client-Verbindungen. Um Ausfallzeiten bei Engine-Upgrades zu minimieren, empfehlen wir Ihnen, die [bewährten Methoden](#)

für [Redis-Clients](#) mit Wiederholungsversuchen und exponentiellem Backoff sowie die bewährten Methoden zur [Minimierung von Ausfallzeiten während der Wartung](#) zu implementieren.

- Sie können nicht direkt von Redis (Clustermodus deaktiviert) auf Redis (Clustermodus aktiviert) aktualisieren, wenn Sie Ihre Engine aktualisieren. Das folgende Verfahren zeigt Ihnen, wie Sie von Redis (Clustermodus deaktiviert) auf Redis (Clustermodus aktiviert) aktualisieren.

So aktualisieren Sie von einer Redis (Clustermodus deaktiviert) auf eine Redis (Clustermodus aktiviert) Engine-Version

1. Erstellen Sie ein Backup Ihres Redis-Clusters (Clustermodus deaktiviert) oder Ihrer Replikationsgruppe. Weitere Informationen finden Sie unter [Erstellen manueller Backups](#).
 2. Verwenden Sie das Backup, um einen Redis-Cluster (Clustermodus aktiviert) mit einem Shard (Knotengruppe) zu erstellen und zu übertragen. 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 [Seeding eines neuen selbst entworfenen Clusters mit einem extern erstellten Backup](#).
 3. Löschen Sie den alten Redis-Cluster (Clustermodus deaktiviert) oder die Replikationsgruppe. Weitere Informationen finden Sie unter [Löschen eines Clusters](#) oder [Löschen einer Replikationsgruppe](#).
 4. Skalieren Sie den neuen Redis-Cluster (Clustermodus aktiviert) oder die Replikationsgruppe auf die Anzahl der Shards (Knotengruppen), die Sie benötigen. Weitere Informationen finden Sie unter [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#).
- 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 Redis-Cluster und Cluster mit deaktivierter Multi-AZ-Funktion sollte Redis ausreichend Speicher zur Verfügung gestellt werden, wie in [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#) beschrieben. In diesen Fällen steht der primäre Knoten während des Upgrade-Prozesses für Serviceanfragen nicht zur Verfügung.
 - Für Redis-Cluster mit aktivierter Multi-AZ-Funktion empfehlen wir, Engine-Upgrades in Zeiträumen mit geringem eingehendem Schreibdatenverkehr zu planen. Beim Upgrade auf Redis 5.0.6 oder höher steht der primäre Cluster während des Upgrade-Prozesses 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.

So führen Sie ein Upgrade von Engine-Versionen aus

Sie initiieren Versionsaktualisierungen für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie sie mithilfe der ElastiCache-Konsole, der AWS CLI oder der ElastiCache-API ändern und eine neuere Engine-Version angeben. Weitere Informationen finden Sie unter den folgenden Themen.

So ändern Sie Cluster und Replikationsgruppen

| Cluster | Replikationsgruppen |
|--|---|
| Mit dem AWS Management Console | Unter Verwendung der AWS Management Console |
| Verwenden Sie den AWS CLI | Mit dem AWS CLI |
| Verwenden der ElastiCache API | Verwenden der API ElastiCache |

Beheben blockierter Redis-Engine-Upgrades

Wie in der folgenden Tabelle dargestellt, wird das Upgrade Ihrer Redis-Engine blockiert, wenn eine Skalierungsoperation aussteht.

| Ausstehende Vorgänge | Blockierte Vorgänge |
|----------------------|------------------------------|
| Aufwärtsskalierung | Unmittelbares Engine-Upgrade |

| Ausstehende Vorgänge | Blockierte Vorgänge |
|---------------------------------------|---------------------------------|
| Engine-Upgrade | Unmittelbares Aufwärtsskalieren |
| Aufwärtsskalierung und Engine-Upgrade | Unmittelbares Aufwärtsskalieren |
| | Unmittelbares Engine-Upgrade |

So lösen Sie ein blockiertes Redis-Engine-Upgrade auf

- Führen Sie eine der folgenden Aktionen aus:
 - Planen Sie Ihr Redis-Engine-Upgrade für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Apply immediately deaktivieren.

Mit der CLI verwenden Sie `--no-apply-immediately`. Mit der API verwenden Sie `ApplyImmediately=false`.
 - Warten Sie bis zum nächsten Wartungsfenster (oder länger), um Ihr Redis-Engine-Upgrade auszuführen.
 - Fügen Sie dieser Cluster-Änderung die Operation zum Aufwärtsskalieren von Redis mit aktiviertem Kontrollkästchen Apply Immediately hinzu.

Mit der CLI verwenden Sie `--apply-immediately`. Mit der API verwenden Sie `ApplyImmediately=true`.

Dieser Ansatz bricht das Engine-Upgrade während des nächsten Wartungsfensters effektiv ab, indem es direkt ausgeführt wird.

Unterstützte ElastiCache-for-Redis-Versionen

ElastiCache-Serverless-Caches unterstützen die folgenden Redis-Versionen:

- [ElastiCache für Redis Version 7.1 \(erweitert\)](#)

Selbst entworfene ElastiCache-Cluster unterstützen die folgenden Redis-Versionen:

- [ElastiCache für Redis Version 7.1 \(erweitert\)](#)
- [ElastiCache für Redis der Version 7.0 \(erweitert\)](#)
- [ElastiCache for Redis Version 6.2 \(erweitert\)](#)
- [ElastiCache for Redis Version 6.0 \(erweitert\)](#)
- [ElastiCache for Redis Version 5.0.6 \(erweitert\)](#)
- [ElastiCache für Redis Version 5.0.5 \(veraltet, Version 5.0.6 verwenden\)](#)
- [ElastiCache für Redis Version 5.0.4 \(veraltet, Version 5.0.6 verwenden\)](#)
- [ElastiCache für Redis Version 5.0.3 \(veraltet, Version 5.0.6 verwenden\)](#)
- [ElastiCache für Redis Version 5.0.0 \(veraltet, Version 5.0.6 verwenden\)](#)
- [ElastiCache for Redis Version 4.0.10 \(erweitert\)](#)
- [Frühere Versionen für das Ende der Lebensdauer \(EOL\) \(3.x\)](#)
- [Frühere Versionen für das Ende der Lebensdauer \(EOL\) \(2.x\)](#)

ElastiCache für Redis 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 wichtige Verbesserungen](#) ein:

Wir haben die erweiterte E/A-Thread-Funktionalität auf die Logik der Präsentationsebene ausgedehnt. Mit Präsentationsschicht sind die erweiterten E/A-Threads gemeint, die jetzt nicht nur Client-Eingaben lesen, sondern die Eingabe auch im Redis-Binärbefehlsformat analysieren. Diese wird dann zur Ausführung an den Haupt-Thread weitergeleitet, was zu einer Leistungssteigerung führt. Verbessertes Redis-Speicherzugriffsmuster. Die Ausführungsschritte vieler Datenstrukturoperationen sind miteinander verwoben, um Speicherzugriff und reduzierte Speicherzugriffslatenz parallel zu gewährleisten. Wenn ElastiCache auf Graviton3-basiertem System

R7g.4xlarge oder höher ausgeführt wird, können Kunden über 1 Million Anfragen pro Sekunde und Knoten erreichen. Mit den Leistungsverbesserungen von ElastiCache für Redis v7.1 können Kunden im Vergleich zu ElastiCache für Redis v7.0 einen um bis zu 100 % höheren Durchsatz und eine um 50 % geringere P99-Latenz erzielen. Diese Verbesserungen sind bei Knotengrößen mit mindestens 8 physischen Kernen (2xlarge auf Graviton und 4xlarge auf x86) unabhängig vom CPU-Typ aktiviert und erfordern keine Client-Änderungen.

Note

ElastiCache v7.1 ist mit OSS Redis v7.0 kompatibel.

ElastiCache für Redis der Version 7.0 (erweitert)

ElastiCache für Redis 7.0 bietet eine Reihe von Verbesserungen und Unterstützung für neue Funktionen:

- [Redis Functions](#) (Redis-Funktionen): ElastiCache für Redis 7 bietet Unterstützung für Redis-Funktionen und eine verwaltete Umgebung, die es Entwicklern ermöglicht, [LUA-Skripts](#) mit Anwendungslogik auszuführen, die im ElastiCache-Cluster gespeichert ist, ohne dass die Clients die Skripts bei jeder Verbindung erneut an den Server senden müssen.
- [ACL](#)-Verbesserungen: ElastiCache für Redis 7 bietet Unterstützung für die nächste Version von Zugriffssteuerungslisten (ACLs). Mit ElastiCache für Redis 7 können Kunden jetzt mehrere Berechtigungsgruppen für bestimmte Schlüssel oder Schlüsselräume in Redis angeben.
- [Sharded Pub/Sub](#): ElastiCache für Redis 7 bietet Unterstützung für die Ausführung von Redis-Pub/Sub-Funktionen mit Sharding, wenn ElastiCache mit aktiviertem Cluster-Modus (CME) ausgeführt wird. Die Pub/Sub-Funktionen von Redis ermöglichen es Herausgebern, Nachrichten an eine beliebige Anzahl von Abonnenten eines Kanals zu senden. Mit Amazon ElastiCache für Redis 7 sind Kanäle an einen Shard im ElastiCache-Cluster gebunden, wodurch die Notwendigkeit entfällt, Kanalinformationen über Shards weiterzuleiten. Das führt zu einer verbesserten Skalierbarkeit.
- Verbessertes E/A-Multiplexing: ElastiCache for Redis Version 7 führt ein verbessertes E/A-Multiplexing ein, das einen höheren Durchsatz und eine geringere Latenz für Workloads mit hohem Durchsatz bietet, die viele gleichzeitige Client-Verbindungen zu einem ElastiCache-Cluster haben. Wenn Sie beispielsweise einen Cluster von r6g.xlarge-Knoten verwenden und 5200 gleichzeitige Clients ausführen, können Sie im Vergleich zu ElastiCache for Redis Version 6 einen um bis zu 72 % höheren Durchsatz (Lese- und Schreibvorgänge pro Sekunde) und eine um bis zu 71 % geringere P99-Latenz erzielen.

Weitere Informationen zur Redis-7.0-Version finden Sie in den [Redis-7.0-Versionshinweisen](#) bei Redis auf GitHub.

ElastiCache for Redis Version 6.2 (erweitert)

ElastiCache for Redis 6.2 enthält Leistungsverbesserungen für TLS-fähige Cluster, die x86-Knotentypen mit 8 vCPUs oder mehr oder Graviton2-Knotentypen mit 4 vCPUs oder mehr verwenden. Diese Optimierungen verbessern den Durchsatz und reduzieren die Zeit für den Aufbau von Clientverbindungen, indem sie die Verschlüsselung auf andere vCPUs auslagern. Mit Redis 6.2 können Sie außerdem den Zugriff auf Pub/Sub-Kanäle mit ACL-Regeln (Access Control List, Zugriffskontrollliste) verwalten.

Mit dieser Version führen wir auch Unterstützung für Daten-Tiering auf Cluster-Knoten ein, die lokal angeschlossene NVMe-SSDs enthalten. Weitere Informationen finden Sie unter [Daten-Tiering](#).

Redis-Engine-Version 6.2.6 unterstützt das native JSON-Format (JavaScript Object Notation), welches eine einfache, schemalose Methode zum Codieren komplexer Datensätze innerhalb von Redis-Clustern ist. Mit der JSON-Unterstützung können Sie die Leistung und die Redis-APIs für Anwendungen nutzen, die über JSON betrieben werden. Weitere Informationen finden Sie unter [Erste Schritte mit JSON](#). Die JSON-bezogenen Metriken `JsonBasedCmds` und `JsonBasedCmdsLatency` werden in CloudWatch integriert, um die Nutzung dieses Datentyps zu überwachen. Weitere Informationen finden Sie unter [Metriken für Redis](#).

Sie geben die Engine-Version mit 6.2 an. ElastiCache for Redis ruft automatisch die bevorzugte verfügbare Patch-Version von Redis 6.2 auf. 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 6.2 gestartet. Die Angabe der Engine-Version 6.x in der API resultiert in der neuesten Nebenversion von Redis 6.

Für vorhandene 6.0-Cluster können Sie sich für das nächste automatische Upgrade einer Nebenversion anmelden, indem Sie den Parameter `AutoMinorVersionUpgrade` in den APIs `CreateCacheCluster`, `ModifyCacheCluster`, `CreateReplicationGroup` oder `ModifyReplicationGroup` auf `yes` festlegen. ElastiCache for Redis 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 Aufruf der `DescribeCacheEngineVersions`-API wird der Parameter `EngineVersion` auf den Wert 6.2 festgelegt und die tatsächliche Engine-Version mit der Patch-Version wird im Feld `CacheEngineVersionDescription` zurückgegeben.

Weitere Informationen zum Redis 6.2-Release finden Sie in den [Redis 6.2-Versionshinweisen](#) bei Redis auf GitHub.

ElastiCache for Redis Version 6.0 (erweitert)

Amazon ElastiCache for Redis stellt die nächste Version der Redis-Engine vor, die die [Authentifizierung von Benutzern mit rollenbasierter Zugriffssteuerung](#), clientseitiges Caching und erhebliche betriebliche Verbesserungen umfasst.

Ab Redis 6.0 bietet ElastiCache for Redis eine einzelne Version für jede Redis OSS-Hauptversion anstelle mehrerer Patch-Versionen an. ElastiCache for Redis verwaltet automatisch die Patch-Version Ihrer laufenden Cache-Cluster, um eine verbesserte Leistung und erhöhte Sicherheit zu gewährleisten.

Sie können sich auch für das nächste automatische Upgrade der Nebenversion anmelden, indem Sie den Parameter `AutoMinorVersionUpgrade` auf `yes` festlegen. ElastiCache for Redis verwaltet das Upgrade der Nebenversion durch Self-Service-Updates. Weitere Informationen finden Sie unter [Service-Updates in ElastiCache](#).

Sie geben die Engine-Version mit `6.0` an. ElastiCache for Redis ruft automatisch die bevorzugte verfügbare Patch-Version von Redis 6.0 auf. 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 6.0 gestartet. Jede Anfrage mit einem bestimmten Patch-Versionswert wird abgelehnt, eine Ausnahme wird ausgelöst und der Vorgang schlägt fehl.

Beim Aufruf der `DescribeCacheEngineVersions`-API wird der Parameter `EngineVersion` auf den Wert `6.0` festgelegt und die tatsächliche Engine-Version mit der Patch-Version wird im Feld `CacheEngineVersionDescription` zurückgegeben.

Weitere Informationen zur Redis 6.0-Version finden Sie in den [Redis 6.0-Versionshinweisen](#) bei Redis auf GitHub.

ElastiCache for Redis Version 5.0.6 (erweitert)

Amazon ElastiCache für Redis führt die nächste Version der Redis-Engine ein, die Fehlerbehebungen 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.
- Funktion zum Umbenennen von Befehlen: Ein Parameter mit der Bezeichnung `rename-commands`, der Ihnen das Umbenennen potenziell gefährlicher oder teurer Redis-Befehle ermöglicht, die zu versehentlichen Datenverlusten führen können, z. B. `FLUSHALL` oder `FLUSHDB`. Dies ähnelt der `rename-command`-Konfiguration in Open-Source-Redis. ElastiCache hat jedoch die Benutzerfreundlichkeit mithilfe 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 [Redis 5.0.3-Parameteränderungen](#).

- **Redis-Streams:** Hierdurch wird eine Protokollstruktur modelliert, mit der neue Elemente in Echtzeit angehängt werden. 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 [Introduction to Redis Streams \(Einführung in Redis-Streams\)](#).
- Unterstützung für eine Reihe von Stream-Befehlen wie etwa `XADD`, `XRANGE` und `XREAD`. Weitere Informationen finden Sie unter [Redis Streams Commands \(Befehle für Redis-Streams\)](#).
- Eine Anzahl neuer und umbenannter Parameter. Weitere Informationen finden Sie unter [Redis 5.0.0-Parameteränderungen](#).
- Eine neue Redis-Metrik, `StreamBasedCmds`.

- Etwas schnellere Snapshot-Zeit für Redis-Knoten

Important

Amazon ElastiCache for Redis hat zwei kritische Fehlerkorrekturen von [Redis-Open-Source-Version 5.0.1](#) zurückportiert. Sie sind im Folgenden aufgeführt:

- RESTORE-Antwort bei fehlender Übereinstimmung, wenn bestimmte Schlüssel bereits abgelaufen sind.
- Der Befehl XCLAIM kann einen falschen Eintrag zurückgeben oder die Synchronisierung des Protokolls verhindern.

Beide Fehlerbehebungen sind in ElastiCache-for-Redis-Unterstützung für Redis Engine-Version 5.0.0 enthalten und werden in zukünftigen Versionsupdates verwendet.

Weitere Informationen finden Sie in den [Versionshinweisen zu Redis 5.0.6](#) unter Redis auf GitHub.

ElastiCache für Redis Version 5.0.5 (veraltet, Version 5.0.6 verwenden)

Amazon ElastiCache für Redis führt die nächste Version der Redis-Engine ein. Es umfasst Online-Konfigurationsänderungen für ElastiCache for Redis von Auto-Failover-Clustern während aller geplanten Operationen. Sie können den Cluster jetzt skalieren, die Redis-Engine-Version aktualisieren und Patches und Wartungsupdates anwenden, während der Cluster online bleibt und weiterhin eingehende Anfragen bedient. Das beinhaltet auch Fehlerbehebungen.

Weitere Informationen finden Sie in den [Redis 5.0.5 Versionshinweisen](#) unter Redis auf GitHub.

ElastiCache für Redis Version 5.0.4 (veraltet, Version 5.0.6 verwenden)

Amazon ElastiCache for Redis führt die nächste Version der Redis-Engine ein, die von Amazon ElastiCache unterstützt wird. 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 [Redis 5.0.4-Versionshinweisen](#) unter Redis auf GitHub.

ElastiCache für Redis Version 5.0.3 (veraltet, Version 5.0.6 verwenden)

Amazon ElastiCache für Redis stellt die nächste Version der Redis-Engine vor, die von Amazon ElastiCache unterstützt wird und Fehlerbehebungen enthält.

ElastiCache für Redis Version 5.0.0 (veraltet, Version 5.0.6 verwenden)

Amazon ElastiCache for Redis stellt die nächste Hauptversion der Redis-Engine vor, die von Amazon ElastiCache unterstützt wird. ElastiCache for Redis 5.0.0 bietet Unterstützung für die folgenden Verbesserungen:

- Redis-Streams: Hierdurch wird eine Protokollstruktur modelliert, mit der neue Elemente in Echtzeit angehängt werden. 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 [Introduction to Redis Streams \(Einführung in Redis-Streams\)](#).
- Unterstützung für eine Reihe von Stream-Befehlen wie etwa XADD, XRANGE und XREAD. Weitere Informationen finden Sie unter [Redis Streams Commands \(Befehle für Redis-Streams\)](#).
- Eine Anzahl neuer und umbenannter Parameter. Weitere Informationen finden Sie unter [Redis 5.0.0-Parameteränderungen](#).
- Eine neue Redis-Metrik, `StreamBasedCmds`.
- Etwas schnellere Snapshot-Zeit für Redis-Knoten

ElastiCache for Redis Version 4.0.10 (erweitert)

Amazon ElastiCache for Redis stellt die nächste Hauptversion der Redis-Engine vor, die von Amazon ElastiCache unterstützt wird. ElastiCache for Redis 4.0.10 unterstützt die folgenden Verbesserungen:

- Sowohl die Größenänderung des Online-Clusters als auch die Verschlüsselung in einer einzigen ElastiCache-for-Redis-Version. Weitere Informationen finden Sie hier:

- [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#)
- [Online-Resharding und Shard-Rebalancing für Redis \(Clustermodus aktiviert\)](#)
- [Datensicherheit in Amazon ElastiCache](#)
- Eine Anzahl neuer Parameter. Weitere Informationen finden Sie unter [Redis 4.0.10-Parameteränderungen](#).
- Unterstützung für eine Reihe von Speicherbefehlen wie etwa MEMORY. Weitere Informationen finden Sie unter [Redis Commands](#) (suchen Sie nach MEMO).
- 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 Flushes und Löschungen. ElastiCache for Redis unterstützt die Ausführung von Befehlen wie UNLINK, FLUSHDB und FLUSHALL in einem anderen Thread als dem Hauptthread. Dies hilft, Leistung und Reaktionszeiten für Ihre Anwendungen zu verbessern, indem Speicher asynchron freigegeben wird.
- Eine neue Redis-Metrik, ActiveDefragHits. Weitere Informationen finden Sie unter [Metriken für Redis](#).

Redis-Benutzer (Clustermodus deaktiviert), die Redis-Version 3.2.10 ausführen, können die Konsole verwenden, um ihre Cluster per Online-Upgrade zu aktualisieren.

Vergleichen von ElastiCache-for-Redis-Unterstützung für Clustergrößenanpassung und Verschlüsselung

| Feature | 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 FedRAMP-, HIPAA- und PCI DSS-konforme Anwendungen. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

Frühere Versionen für das Ende der Lebensdauer (EOL) (3.x)

ElastiCache for Redis Version 3.2.10 (erweitert)

Amazon ElastiCache for Redis stellt die nächste Hauptversion der Redis-Engine vor, die von Amazon ElastiCache unterstützt wird. ElastiCache for Redis 3.2.10 führt die Online-Cluster-Größenänderung ein, um Shards zum Cluster hinzuzufügen oder daraus zu entfernen, während es weiterhin eingehende E/A-Anfragen verarbeitet. Benutzer von ElastiCache for Redis 3.2.10 verfügen über alle Funktionen früherer Redis-Versionen, mit Ausnahme der Möglichkeit, ihre Daten zu verschlüsseln. Diese Möglichkeit ist derzeit nur in Version 3.2.6 verfügbar.

Vergleichen der ElastiCache-for-Redis-Versionen 3.2.6 und 3.2.10

| Feature | 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 FedRAMP-, HIPAA- und PCI DSS-konforme Anwendungen. Weitere Informationen finden Sie unter [Konformitätsvalidierung für Amazon ElastiCache](#).

Weitere Informationen finden Sie hier:

- [Online-Resharding und Shard-Rebalancing für Redis \(Clustermodus aktiviert\)](#)
- [Online-Größenanpassung von Clustern](#)

ElastiCache for Redis Version 3.2.6 (erweitert)

Amazon ElastiCache for Redis stellt die nächste Hauptversion der Redis-Engine vor, die von Amazon ElastiCache unterstützt wird. Benutzer von ElastiCache for Redis 3.2.6 verfügen über alle Funktionen

früherer 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 for Redis Version 3.2.4 (erweitert)

Amazon ElastiCache for Redis Version 3.2.4 führt die nächste Hauptversion der Redis-Engine ein, die von Amazon ElastiCache unterstützt wird. ElastiCache for Redis 3.2.4-Benutzern stehen alle Funktionen früherer Redis-Versionen zur Verfügung sowie die Option, im Cluster-Modus oder Nicht-Cluster-Modus zu laufen. Die folgende Tabelle enthält eine Zusammenfassung.

Vergleich von Redis 3.2.4 Nicht-Cluster-Modus und Cluster-Modus

| Feature | 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.
- Geoindizierung – Redis 3.2.4 bietet Unterstützung für die Geoindizierung über sechs GEO-Befehle. Weitere Informationen finden Sie in der Dokumentation zu Redis-GEO*-Befehlen [Redis-Befehle: GEO](#) auf der Redis-Befehlsseite (nach GEO gefiltert).

Informationen über zusätzliche Redis 3-Funktionen finden Sie in den [Redis 3.2 Release Notes](#) und [Redis 3.0 Release Notes](#).

Derzeit unterstützt von ElastiCache verwaltetes Redis (Clustermodus aktiviert) die folgenden Redis 3.2-Funktionen nicht:

- Replikatmigration
- Cluster-Ausgleich
- Lua Debugger

ElastiCache deaktiviert die folgenden 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`

Weitere Informationen über die Parameter von Redis 3.2.4 finden Sie unter [Redis 3.2.4-Parameteränderungen](#).

Frühere Versionen für das Ende der Lebensdauer (EOL) (2.x)

ElastiCache for Redis Version 2.8.24 (erweitert)

Zu den Verbesserungen seit Redis Version 2.8.23 gehören Fehlerbehebungen und Protokollierung von Adressen ungültiger Speicherzugriffe. Weitere Informationen finden Sie unter [Redis 2.8 Versionshinweise](#).

ElastiCache for Redis Version 2.8.23 (erweitert)

Zu den Verbesserungen seit Redis Version 2.8.22 gehören Fehlerbehebungen. Weitere Informationen finden Sie unter [Redis 2.8 Versionshinweise](#). 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 Redis 2.8.23-Parametern finden Sie unter [In Redis 2.8.23 \(Enhanced\) hinzugefügte Parameter](#) im ElastiCache-Benutzerhandbuch.

ElastiCache for Redis Version 2.8.22 (erweitert)

Es wurden folgende Redis-Verbesserungen seit Version 2.8.21 eingeführt:

- 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.
- Unterstützung 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 mithilfe von CloudWatch-Metriken](#).

- Eine Reihe entscheidender Fehlerbehebungen im PSYNC-Replikationsverhalten. Weitere Informationen finden Sie unter [Redis 2.8 Versionshinweise](#).
- Um eine verbesserte Replikationsleistung in Multi-AZ-Replikationsgruppen beizubehalten und die Clusterstabilität zu erhöhen, werden Nicht-ElastiCache-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.
- Die Redis-Konfigurationsvariablen `appendonly` und `appendfsync` werden auf der Redis-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 häufig bei PUBSUB-Clients auf.

ElastiCache for Redis Version 2.8.21

Zu den Verbesserungen seit Redis Version 2.8.19 gehören eine Reihe von Fehlerbehebungen. Weitere Informationen finden Sie unter [Redis 2.8 Versionshinweise](#).

ElastiCache for Redis Version 2.8.19

Es wurden folgende Redis-Verbesserungen seit Version 2.8.6 eingeführt:

- Support für HyperLogLog. Weitere Informationen finden Sie unter [Die neue Datenstruktur von Redis: das HyperLogLog](#).
- Der Datentyp „Sorted Set“ verfügt mit den neuen Befehlen `ZRANGEBYLEX`, `ZLEXCOUNT` und `ZREMRANGEBYLEX` nun über Support für lexikografische Bereichsabfragen.
- Damit ein primärer Knoten keine veralteten Daten an Replica-Knoten sendet, schlägt die Master-Synchronisierung fehl, wenn ein untergeordneter Speicherprozess im Hintergrund (`bgsave`) abgebrochen wird.
- Support für `HyperLogLogBasedCommands` CloudWatch Metrik. Weitere Informationen finden Sie unter [Metriken für Redis](#).

ElastiCache for Redis Version 2.8.6

Es wurden folgende Redis-Verbesserungen seit Version 2.6.13 eingeführt:

- 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 for Redis Version 2.6.13

Redis-Version 2.6.13 war die erste Version von Redis, die von Amazon ElastiCache for Redis unterstützt wurde. Multi-AZ wird unter Redis 2.6.13 nicht unterstützt.

Zeitplan für das Lebensende der Redis-Versionen



In diesem Abschnitt werden die Termine für das Ende der Lebensdauer (EOL) für ältere Hauptversionen so festgelegt, wie sie angekündigt werden. Auf diese Weise können Sie Versions- und Upgrade-Entscheidungen für die Zukunft treffen.

Note

Die Patchversionen von ElastiCache für Redis von 5.0.0 bis 5.0.5 sind veraltet. Verwenden Sie Version 5.0.6 oder höher.

Die folgende Tabelle fasst jede Version und ihr angekündigtes EOL-Datum sowie die empfohlene Upgrade-Zielversion zusammen.

Vergangenes EOL

| C f H ic | Quelle für Unterversionen | Empfohlenes Upgrade-Ziel | EOL-Datum |
|-------------------|--|--|-----------------|
| V 3 | 3.2.4, 3.2.6 und 3.2.10 | Version 6.2 oder höher <div data-bbox="613 464 1040 877"><p> Note</p><p>Für die Regionen US-ISO-EAST-1, US-ISO-WEST-1 und US-ISOB-EAST-1 empfehlen wir 5.0.6 oder höher.</p></div> | 31. Juli 2023 |
| V 2 | 2.8,24, 2.8,23, 2.8,22, 2.8,21, 2.8,19, 2.8,12, 2.8,6, 2.6,13 | Version 6.2 oder höher <div data-bbox="613 993 1040 1407"><p> Note</p><p>Für die Regionen US-ISO-EAST-1, US-ISO-WEST-1 und US-ISOB-EAST-1 empfehlen wir 5.0.6 oder höher.</p></div> | 13. Januar 2023 |

So führen Sie ein Upgrade von Engine-Versionen aus

Sie initiieren Versionsaktualisierungen für Ihren Cluster oder Ihre Replikationsgruppe, indem Sie sie mithilfe der ElastiCache-Konsole, der AWS CLI oder der ElastiCache-API ändern und eine neuere Engine-Version angeben. Weitere Informationen finden Sie unter den folgenden Themen.

| So ändern Sie Cluster und Replikationsgruppen | |
|--|---|
| Cluster | Replikationsgruppen |
| Mit dem AWS Management Console | Unter Verwendung der AWS Management Console |
| Verwenden Sie den AWS CLI | Mit dem AWS CLI |
| Verwenden der ElastiCache API | Verwenden der API ElastiCache |

Beheben blockierter Redis-Engine-Upgrades

Wie in der folgenden Tabelle dargestellt, wird das Upgrade Ihrer Redis-Engine blockiert, wenn eine Skalierungsoperation aussteht.

| Ausstehende Vorgänge | Blockierte Vorgänge |
|---------------------------------------|---------------------------------|
| Aufwärtsskalierung | Unmittelbares Engine-Upgrade |
| Engine-Upgrade | Unmittelbares Aufwärtsskalieren |
| Aufwärtsskalierung und Engine-Upgrade | Unmittelbares Aufwärtsskalieren |
| | Unmittelbares Engine-Upgrade |

So lösen Sie ein blockiertes Redis-Engine-Upgrade auf

- Führen Sie eine der folgenden Aktionen aus:
 - Planen Sie Ihr Redis-Engine-Upgrade für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Apply immediately deaktivieren.

Mit der CLI verwenden Sie `--no-apply-immediately`. Mit der API verwenden Sie `ApplyImmediately=false`.

- Warten Sie bis zum nächsten Wartungsfenster (oder länger), um Ihr Redis-Engine-Upgrade auszuführen.
- Fügen Sie dieser Cluster-Änderung die Operation zum Aufwärtsskalieren von Redis mit aktiviertem Kontrollkästchen `Apply Immediately` hinzu.

Mit der CLI verwenden Sie `--apply-immediately`. Mit der API verwenden Sie `ApplyImmediately=true`.

Dieser Ansatz bricht das Engine-Upgrade während des nächsten Wartungsfensters effektiv ab, indem es direkt ausgeführt wird.

Wesentliche Versionsverhalten und Kompatibilitätsunterschiede

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 Redis-Version auf die neueste verfügbare Redis-Version aktualisieren, ohne dass sequentielle Upgrades erforderlich sind. Sie können beispielsweise direkt von Redis Version 3.0 auf Version 7.0 aktualisieren.

Redis-Versionen werden mit einer semantischen Version identifiziert, die eine MAJOR-, MINOR- und PATCH-Komponente umfasst. In Redis 4.0.10 ist beispielsweise die Hauptversion 4, die Unterversion 0 und die Patch-Version ist 10. Diese Werte werden im Allgemeinen basierend auf den folgenden Konventionen schrittweise erhöht:

- HAUPTVERSIONEN stehen für API-inkompatible Änderungen
- Unterversionen stehen für neue Funktionen, die abwärtskompatibel hinzugefügt wurden
- PATCH-Versionen stehen für abwärtskompatible Fehlerbehebungen und nicht funktionale Änderungen

Wir empfehlen, innerhalb einer bestimmten HAUPT-/UNTERVERSION immer auf der neuesten Patch-Version zu bleiben, um die neuesten Leistungs- und Stabilitätsverbesserungen zu erhalten. Ab Redis 6.0 bietet ElastiCache für Redis eine einzelne Version für jede Redis-OSS-Nebenversion anstelle mehrerer Patch-Versionen an. ElastiCache für Redis verwaltet automatisch die Patch-Version Ihrer laufenden Cache-Cluster, um eine verbesserte Leistung und erhöhte Sicherheit zu gewährleisten.

Wir empfehlen außerdem, regelmäßig auf die neueste Hauptversion zu aktualisieren, da die meisten wichtigen Verbesserungen nicht auf ältere Versionen zurückportiert werden. Da ElastiCache die Verfügbarkeit auf eine neue AWS-Region erweitert, unterstützt ElastiCache für Redis die zwei jeweils neuesten MAJOR.MINOR-Versionen für die neue Region. Wenn beispielsweise eine neue AWS-Region gestartet wird und die neuesten MAJOR.MINOR-Versionen von ElastiCache für Redis 7.0 und 6.2 sind, unterstützt ElastiCache für Redis die Versionen 7.0 und 6.2 in der neuen AWS-Region. Wenn neuere MAJOR.MINOR-Versionen von ElastiCache für Redis veröffentlicht werden, wird ElastiCache weiterhin Unterstützung für die neu veröffentlichten Versionen von ElastiCache für Redis hinzufügen. Weitere Informationen zur Auswahl von Regionen für ElastiCache finden Sie unter [Auswahl von Regionen und Availability Zones](#).

Wenn Sie ein Upgrade durchführen, das Haupt- oder Unterversionen umfasst, beachten Sie bitte die folgende Liste, die Verhalten und abwärtskompatible Änderungen enthält, die mit Redis im Laufe der Zeit veröffentlicht wurden.

Redis 7.0 – Verhalten und abwärtskompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie unter [Redis-7.0-Versionshinweise](#).

- `SCRIPT LOAD` und `SCRIPT FLUSH` werden nicht mehr auf Replikate weitergeleitet. Wenn Sie bei Skripten eine gewisse Dauerhaftigkeit benötigen, empfehlen wir Ihnen die Verwendung von [Redis functions](#) (Redis-Funktionen) in Betracht zu ziehen.
- Pubsub-Kanäle sind jetzt für neue ACL-Benutzer standardmäßig 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.

Redis 6.2 – Verhalten und abwärtskompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie unter [Redis 6.2-Versionshinweise](#).

- Die ACL-Flags der Befehle TIME, ECHO, ROLE und LASTSAVE 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 6.0 wird die Reihenfolge der Schlüssel/Wert-Paare, die von einer Map-Antwort auf ein Lua-Skript zurückgegeben werden, geändert. Wenn Ihre Skripte `redis.setresp()` verwenden oder eine Karte zurückgeben (neu in Redis 6.0), bedenken Sie, welche Auswirkungen das Skript bei Upgrades haben kann.

Redis 6.0 – Verhalten und abwärtskompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie unter [Redis 6.0-Versionshinweise](#).

- 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.
- Legen Sie `AutoMinorVersionUpgrade` auf „yes“ fest. ElastiCache for Redis verwaltet das Upgrade der Unterversion durch 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 in Amazon ElastiCache](#).

Redis 5.0 – Verhalten und abwärtskompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie unter [Redis 5.0-Versionshinweise](#).

- 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, das nur in ElastiCache for Redis 5.0 verfügbar ist.
- Wenn Sie ein Upgrade von Redis 4.0 durchführen, geben einige Befehle in LUA-Skripten Argumente in einer anderen Reihenfolge als in früheren Versionen zurück. In Redis 4.0 würde Redis einige Antworten lexographisch anordnen, um die Antworten deterministisch zu machen. Diese Reihenfolge wird nicht angewendet, wenn Skripte durch Effekte repliziert werden.
- In Redis 5.0.3 und höher wird ElastiCache for Redis einige E/A-Arbeiten bei Instance-Typen mit mehr als 4 vCPUs auf Hintergrundkerne auslagern. Dies kann die Leistungseigenschaften von Redis 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.

Redis 4.0 – Verhalten und abwärtskompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie unter [Redis 4.0-Versionshinweise](#).

- 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.

Vergangenes EOL

Redis 3.2 – Verhalten und abwärtskompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie unter [Redis 3.2 Versionshinweise](#).

- Für diese Version sind keine Kompatibilitätsänderungen erforderlich.

Weitere Informationen finden Sie unter [Zeitplan für das Lebensende der Redis-Versionen](#).

Redis 2.8 – Verhalten und abwärtskompatible Änderungen

Eine vollständige Liste der Änderungen finden Sie unter [Redis 2.8 Versionshinweise](#).

- Beginnend mit Redis 2.8.22 wird Redis AOF in ElastiCache for Redis nicht mehr unterstützt. Wir empfehlen die Verwendung von MemoryDB, wenn Daten dauerhaft gespeichert werden müssen.
- Ab Redis 2.8.22 unterstützt ElastiCache for Redis das Anhängen von Replikaten an primäre Replikate, die in ElastiCache gehostet werden, nicht mehr. 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.
- 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 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 Lebensende der Redis-Versionen](#).

ElastiCache Best Practices 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

- [Arbeiten mit Redis](#)
- [Bewährte Methoden für Redis-Clients](#)
- [Verwalten von reserviertem Speicher](#)
- [Bewährte Methoden bei der Arbeit mit selbst entworfenen Clustern](#)
- [Bewährte Methoden für Redis](#)
- [Caching-Strategien](#)

Arbeiten mit Redis

Im Folgenden finden Sie Informationen über die Redis-Schnittstelle in ElastiCache.

Themen

- [Unterstützte und eingeschränkte Redis-Befehle](#)
- [Redis-Konfiguration und -Limits](#)

Unterstützte und eingeschränkte Redis-Befehle

Unterstützte Redis-Befehle

Unterstützte Redis-Befehle

Die folgenden Redis-Befehle werden von Serverless-Caches unterstützt. Zusätzlich zu diesen Befehlen werden außerdem diese [Unterstützte Redis-JSON-Befehle](#) 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

Aktiviert schreibgeschützte Abfragen für eine Verbindung mit einem Redis-Cluster-Replikatknoten.

[Weitere Informationen](#)

- READWRITE

Aktiviert Lese-/Schreibabfragen für eine Verbindung mit einem Redis-Cluster-Replikatknoten.

[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

Führt einen Handshake mit dem Redis-Server durch.

[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

Frägt 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 Sets zurück, die von den HyperLogLog-Schlüsseln beobachtet wurden.

[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

PUBSUB-Befehle verwenden intern Sharded PUBSUB, 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 SHA1-Digest aus.

[Weitere Informationen](#)

- EVALSHA_RO

Führt ein schreibgeschütztes serverseitiges Lua-Skript per SHA1-Digest aus.

[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 ACL-Benutzern verwendet werden kann.

[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 Computerkunst und die 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 Element von einem Set in ein anderes.

[Weitere Informationen](#)

- SPOP

Gibt ein oder mehrere zufällige Elemente 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 Element 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 Elemente 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 Sete 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 ID-Bereichs zurück.

[Weitere Informationen](#)

- XREAD

Gibt Nachrichten aus mehreren Streams zurück, deren IDs größer als die angeforderten sind. 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 ID-Bereichs 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](#)

Beschränkte Redis-Befehle

Um eine verwaltete Serviceerfahrung zu bieten, beschränkt ElastiCache den Zugriff auf bestimmte Cache-Engine-spezifische Befehle, die erweiterte Rechte erfordern. 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`
- `punsubscribe`
- `pubsub numpat`
- `punsubscribe`
- `script kill`
- `slowlog`
- `slowlog get`
- `slowlog help`
- `slowlog len`
- `slowlog reset`
- `swapdb`
- `unwatch`
- `wait`
- `watch`

Redis-Konfiguration und -Limits

Die Redis-Engine bietet eine Reihe von Konfigurationsparametern, von denen einige in ElastiCache für Redis geändert, andere hingegen nicht geändert werden können, um eine stabile Leistung und Zuverlässigkeit zu gewährleisten.

Serverless-Caches

Für Serverless-Caches werden keine Parametergruppen verwendet und die gesamte Redis-Konfiguration kann nicht geändert werden. Die folgenden Redis-Parameter sind vorhanden:

| Name | Details | Beschreibung |
|-------------------------------------|------------------------------------|---|
| acl-pubsub-default | allchannels | Standardberechtigungen für Pub/Sub-Kanäle für ACL-Benutzer im Cache. |
| client-output-buffer-limit | normal 0 0 0 pubsub 32mb 8mb 60 | Für normale Clients besteht kein Pufferlimit. PUB/SUB-Clients werden getrennt, wenn sie den 32-MiB-Backlog oder den 8-MiB-Backlog für 60 Sekunden unterbrechen. |
| 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 Cluster-Modus aktiviert, sodass sie ihre Daten transparen |

| Name | Details | Beschreibung |
|--|--------------|---|
| | | nt auf mehrere Backend-Shards partitionieren können. Alle Slots werden den Clients so angezeigt, als gehörten sie einem einzigen virtuellen Knoten. |
| <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 | Die maximale Ausführungszeit für ein Lua-Skript in Millisekunden, bevor ElastiCache Maßnahmen ergreift, um das Skript zu stoppen. Wenn <code>lua-time-limit</code> überschritten wird, geben alle Redis-Befehle einen Fehler im Format <code>___-BUSY</code> zurück. Da dieser Zustand viele wichtige Redis-Operationen stören kann, gibt ElastiCache zuerst einen <code>SCRIPT KILL</code> -Befehl aus. Wenn dies nicht erfolgreich ist, erzwingt ElastiCache einen Neustart von Redis. |
| <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> | volatile-lru | Elemente mit einem TTL-Set werden nach der Schätzung der ältesten verwendeten Objekte (LRU) bereinigt, wenn das Speicherlimit eines Caches erreicht ist. |

| Name | Details | Beschreibung |
|------------------------|---------------------------------------|---|
| notify-keyspace-events | (eine leere Zeichenfolge) | Keyspace-Ereignisse werden in Serverless-Caches derzeit nicht unterstützt. |
| port | Primärer Port: 6379 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 Redis-Schlüssel- oder Kanalnamen. Clients, die auf Schlüssel verweisen, die größer sind, wird eine Fehlermeldung angezeigt. |
| Größe des Lua-Skripts | 4 MiB | Die maximale Größe eines einzelnen Redis-Lua-Skripts. Versuche, ein Lua-Skript zu laden, das größer ist als dieses, führen zu einer Fehlermeldung. |

| Name | Details | Beschreibung |
|------------|---------|--|
| Slot-Größe | 32 GiB | Die maximale Größe eines einzelnen Redis-Hash-Slots. Clients, die versuchen, mehr Daten als diesen Wert in einem einzelnen Redis-Slot zu speichern, lösen die Bereinigungsrichtlinie für den Slot aus. Wenn keine Schlüssel bereinigt werden können, wird ein Fehler wegen Speichermangel (OOM) angezeigt. |

Selbst entworfene Cluster

Informationen über Standardwerte der Konfigurationsparameter und die konfigurierbaren Werte für selbst entworfene Cluster finden Sie unter [Redis-spezifische Parameter](#). Die Standardwerte werden generell empfohlen, es sei denn, Sie haben einen bestimmten Anwendungsfall, bei dem sie außer Kraft gesetzt werden müssen.

Bewährte Methoden für Redis-Clients

Lernen Sie bewährte Methoden für gängige Szenarien kennen und folgen Sie den Codebeispielen einiger der beliebtesten Open-Source-Client-Bibliotheken von Redis (redis-py, PHPRedis und Lettuce).

Themen

- [Hohe Anzahl von Verbindungen](#)
- [Erkennung von Redis-Cluster-Clients und exponentielles Backoff](#)
- [Konfigurieren eines clientseitigen Timeouts](#)
- [Konfigurieren eines serverseitigen Leerlauf-Timeouts](#)
- [Redis-Lua-Skripte](#)
- [Speichern großer zusammengesetzter Elemente](#)
- [Lettuce-Client-Konfiguration](#)
- [Beispiele für IPv6-Clients](#)

Hohe Anzahl von Verbindungen

Serverless-Caches und einzelne Knoten von ElastiCache für Redis unterstützen bis zu 65 000 gleichzeitige Client-Verbindungen. Zur Optimierung der Leistung empfehlen wir jedoch, dass Client-Anwendungen nicht ständig mit diesem Verbindungsaufkommen arbeiten. Redis ist ein Single-Thread-Prozess, der auf einer Ereignisschleife basiert, bei der eingehende Client-Anfragen sequenziell 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 dem Redis-Server zu vermeiden:

- Führen Sie Lesevorgänge von Read Replicas aus durch. Verwenden Sie dazu die ElastiCache-Reader-Endpunkte im deaktivierten Cluster-Modus oder Replikate für Lesevorgänge im aktivierten Cluster-Modus, einschließlich eines Serverless-Caches.
- Verteilen Sie den Schreibdatenverkehr auf mehrere Primärknoten. Es gibt zwei Methoden dafür: Sie können einen Redis-Cluster mit mehreren Shards mit einem Client verwenden, der den Redis-Cluster-Modus 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 Redis-Befehlen ein rechenintensiver Vorgang. Beispielsweise ist die Bearbeitung einer SET/GET-Anfrage 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 von PHPRedis zeigt, dass für jede neue Benutzeranforderung eine neue Verbindung hergestellt 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 (Amazon EC2)-Instance verglichen, die mit einem Graviton2 (m6g.2xlarge)-Knoten von ElastiCache für Redis verbunden ist. 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);
}
```

Erkennung von Redis-Cluster-Clients und exponentielles Backoff

Wenn Sie im aktivierten Cluster-Modus eine Verbindung mit einem Cluster von ElastiCache für Redis herstellen, muss die entsprechende Redis-Client-Bibliothek 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. im Fall eines Failovers, bei dem alle vom ehemaligen Primärknoten bereitgestellten Slots vom Replikat übernommen werden, oder beim Re-Sharding, wenn Slots vom primären Quellknoten zum primären Zielknoten verschoben werden.

Die Client-Erkennung erfolgt in der Regel durch Ausgabe eines CLUSTER SLOT- oder CLUSTER NODE-Befehls an den Redis-Server. Wir empfehlen die CLUSTER SLOT-Methode, da sie das Set der Slot-Bereiche und die zugehörigen Primär- und Replikatknoten an den Client zurückgibt. Dies erfordert kein zusätzliches Parsen durch den Client und ist effizienter.

Abhängig von der Cluster-Topologie kann die Größe der Antwort auf den Befehl CLUSTER SLOT 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 Redis-Server bei einer CPU-Auslastung von 100 % über einen längeren Zeitraum nicht reagiert. Der Ausfall verlängert sich, wenn jeder CLUSTER SLOT-Befehl eine große Anzahl von Knoten im Cluster-Bus verarbeiten muss. Aufgrund dieses Verhaltens haben wir in der Vergangenheit mehrere Client-Ausfälle in verschiedenen 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 ermitteln](#), um den Cluster-Endpunkt 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, PHPRedis und Lettuce.

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 `retry_on_timeout`-Argument aktiviert werden, das beim Erstellen eines [Redis](#)-Objekts angegeben wird. 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 eines exponentiellen Backoffs in [PHPRedis \(#1986\)](#) gesendet, die inzwischen zusammengeführt und [dokumentiert](#) wurde. Für diejenigen, die die neueste Version von PHPRedis verwenden, ist eine manuelle Implementierung nicht erforderlich, aber wir haben hier die Referenz für diejenigen angegeben, die frühere Versionen verwenden. 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 eines 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 Redis-Befehle können rechenintensiver sein als andere. Dazu gehören zum Beispiel Lua-Skripte oder MULTI/EXEC-Transaktionen, die mehrere

Befehle enthalten, welche 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-Transaktionen oder Lua-Skripten, die aus mehreren einzelnen Redis-Befehlen bestehen
- Lesen von großen Werten
- Durchführen von Blockiervorgängen wie BLPOP

Im Falle eines Blockiervorgangs wie BLPOP empfiehlt es sich, das Befehls-Timeout auf einen Wert festzulegen, der unter dem Socket-Timeout liegt.

Im Folgenden finden Sie Codebeispiele für die Implementierung eines clientseitigen Timeouts in redis-py, PHPRedis und Lettuce.

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

Im Folgenden finden Sie ein Codebeispiel für 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"); // BLPOP with 1 second timeout
    } finally {
        if (connection != null) {
            connection.close();
        }

        if (client != null){
            client.shutdown();
        }
    }
}

```

```
}
```

Konfigurieren eines serverseitigen Leerlauf-Timeouts

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 [Redis-spezifische Parameter](#). 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.

Redis-Lua-Skripte

Redis unterstützt 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 Redis auswirken können.

Nicht parametrisierte Lua-Skripte

Jedes Lua-Skript wird auf dem Redis-Server zwischengespeichert, bevor es ausgeführt wird. Nicht parametrisierte Lua-Skripte sind einzigartig, was dazu führen kann, dass der Redis-Server eine große Anzahl von Lua-Skripten speichert und mehr Speicher verbraucht. Damit dies vermieden wird, stellen Sie sicher, dass alle Lua-Skripte parametrisiert sind, und führen Sie bei Bedarf regelmäßig SCRIPT FLUSH aus, um zwischengespeicherte Lua-Skripte zu bereinigen.

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 regulären Redis-Befehl. 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 lang andauerndes Lua-Skript kann dazu führen, dass der 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 einige Möglichkeiten, wie ein Lua-Skript weiterhin neue Daten in Redis schreiben kann, auch wenn Redis den Wert `maxmemory` überschritten hat.

- Das Skript startet, wenn der Redis-Server unter `maxmemory` liegt, und enthält mehrere Schreibvorgänge
- Der erste Schreibbefehl des Skripts verbraucht keinen Speicher (wie `DEL`), gefolgt von weiteren Schreibvorgängen, die Speicher belegen
- Sie können dieses Problem verringern, indem Sie im Redis-Server eine andere geeignete Bereinigungsrichtlinie als `noeviction` konfigurieren. Auf diese Weise kann Redis Elemente bereinigen und Speicherplatz zwischen Lua-Skripten freigeben.

Speichern großer zusammengesetzter Elemente

In einigen Szenarien kann eine Anwendung große zusammengesetzte Elemente in Redis speichern (z. B. einen Hash-Datensatz mit mehreren GB). Diese Vorgehensweise wird nicht empfohlen, da sie häufig zu Leistungsproblemen bei Redis führt. Der Client kann beispielsweise den Befehl `HGETALL` ausführen, um die gesamte Hash-Sammlung mit mehreren GB abzurufen. Dies kann zu erheblichem Speicherdruck auf den Redis-Server führen, der das große Element im Client-Ausgabepuffer zwischenspeichert. Außerdem migriert ElastiCache bei der Slot-Migration im Cluster-Modus keine Slots, 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 Befehl MGET 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- oder HMGET-Befehlen anstelle von HGETALL für Hash-Sammlungen.

Lettuce-Client-Konfiguration

In diesem Abschnitt werden die empfohlenen Konfigurationsoptionen für Java und Lettuce sowie deren Anwendung für ElastiCache-Cluster beschrieben.

Die Empfehlungen in diesem Abschnitt wurden mit Lettuce Version 6.2.2 getestet.

Themen

- [Beispiel: Lettuce-Konfiguration für Cluster-Modus und aktiviertem TLS](#)
- [Beispiel: Lettuce-Konfiguration für deaktivierten Cluster-Modus und aktiviertes TLS](#)

Java-DNS-Cache-TTLs

Die Java Virtual Machine (JVM) speichert DNS-Namensauflösungen zwischen. Wenn die JVM einen Hostnamen zu einer IP-Adresse auflöst, speichert sie die IP-Adresse für einen bestimmten Zeitraum zwischen. Diese Zeit ist als Time-to-Live (TTL, Lebensdauer) bekannt.

Die Wahl des TTL-Werts ist ein Kompromiss zwischen Latenz auf der einen und Reaktionsfähigkeit auf Änderungen auf der anderen Seite. Bei kürzeren TTLs bemerken DNS-Resolver Aktualisierungen im DNS des Clusters schneller. Dadurch kann Ihre Anwendung schneller auf Ersetzungen oder andere Workflows reagieren, denen Ihr Cluster unterzogen wird. Ist die TTL jedoch zu niedrig, erhöht sich das Abfragevolumen, was die Latenz Ihrer Anwendung erhöhen kann. Zwar gibt es keinen korrekten TTL-Wert, ist es bei der Festlegung des TTL-Werts beachtenswert, zu berücksichtigen, wie lange Sie darauf warten können, dass eine Änderung wirksam wird.

Da ElastiCache-Knoten DNS-Namenseinträge nutzen, die sich möglicherweise ändern, empfehlen wir, dass Sie Ihre JVM mit einem niedrigen TTL-Wert von 5 bis 10 Sekunden konfigurieren. Auf diese

Weise wird bei Änderung der IP-Adresse eines Knotens sichergestellt, dass Ihre Anwendung die neue IP-Adresse der Ressource durch erneute Abfrage des DNS-Eintrags abrufen und nutzen kann.

Bei einigen Java-Konfigurationen ist die JVM-Standard-TTL so festgelegt, dass DNS-Einträge nie aktualisiert werden, bis die JVM neu gestartet wird.

Einzelheiten zum Festlegen der JVM-TTL finden Sie unter [So legen Sie die 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 für diesen URI gibt eine Liste aller verfügbaren Knoten im Cluster zurück und wird während der Cluster-Initialisierung nach dem Zufallsprinzip auf einen davon aufgelöst. Weitere Informationen zur Funktionsweise der Topologieaktualisierung finden Sie weiter unten in diesem Thema unter `dynamicRefreshResources`.

SocketOption

Aktivieren Sie [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: Cluster-Optionen „Cluster-Modus aktiviert“

Aktivieren Sie [AutoReconnect](#), wenn die Verbindung unterbrochen wird.

Legen Sie [CommandTimeout](#) fest. Weitere Einzelheiten finden Sie im Abschnitt „Timeouts“ weiter unten in diesem Thema.

Legen Sie [nodeFilter](#) fest, um ausgefallene Knoten aus der Topologie herauszufiltern. Lettuce speichert alle Knoten, die sich in der Ausgabe „Cluster-Knoten“ befinden (einschließlich der Knoten mit PFAIL/FAIL-Status), in den „Partitionen“ des Clients (auch als Shards bekannt). 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 das `clusterTopologie`-Element aktualisiert wird, weist die Zuordnung der Cluster-Busknoten einen kurzen Zeitraum auf, in dem der ausgefallene Knoten als `FAIL`-Knoten aufgeführt wird, bevor er vollständig aus der Topologie entfernt wird. Während dieser Zeit betrachtet der Redis-Client von Lettuce ihn als intakten Knoten und stellt kontinuierlich eine Verbindung damit her. Dies führt zu einem Fehler, nachdem alle Wiederholungsversuche durchgeführt wurden.

Beispiel:

```
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 `DynamicRefreshSources` auf „wahr“ festgelegt 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. Das Vorhandensein mehrerer Seed-Knoten (wenn `DynamicRefreshSources` auf „wahr“ festgelegt ist) verringert die Wahrscheinlichkeit dieses Problems, da zumindest einige der Seed-Knoten nach einem Failover mit dem neu heraufgestuften Primärknoten über eine aktualisierte Topologieansicht verfügen sollten.

`ClusterTopologyRefreshOptions`: Optionen zur Steuerung der Aktualisierung der Cluster-Topologie des Clients mit aktiviertem Cluster-Modus

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 mit ElastiCache deaktivierte Cluster-Modus ist nicht mit `MasterSlaveTopologyRefresh` von Lettuce kompatibel. 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 [Suche nach den Endpunkten eines Redis-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

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.

Aktivieren Sie [enablePeriodicRefresh](#). Dies ermöglicht eine regelmäßige Aktualisierung der Cluster-Topologie, sodass der Client die Cluster-Topologie in den Intervallen des `refreshPeriod`-Elements aktualisiert (Standard: 60 Sekunden). 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 Sie [enableAllAdaptiveRefreshTriggers](#). Dies ermöglicht eine adaptive Aktualisierung der Topologie, die alle [Trigger](#) verwendet: `MOVED_REDIRECT`, `ASK_REDIRECT`, `PERSISTENT_RECONNECTS`, `UNCOVERED_SLOT`, `UNKNOWN_NODE`. Adaptive Aktualisierungs-Triggers initiieren Aktualisierungen der Topologieansicht auf der Grundlage von Ereignissen, die während des Betriebs des Redis-Clusters 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).

Aktivieren Sie [closeStaleConnections](#). Dadurch können alte Verbindungen beim Aktualisieren der Cluster-Topologie geschlossen werden. Das Element wird nur wirksam, wenn [ClusterTopologyRefreshOptions.isPeriodicRefreshEnabled\(\)](#) wahr ist. 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.

Aktivieren Sie [dynamicRefreshResources](#). Wir empfehlen, das `dynamicRefreshResources`-Element für kleine Cluster zu aktivieren und es für große Cluster zu deaktivieren. Mit dem `dynamicRefreshResources`-Element können Sie Cluster-Knoten vom bereitgestellten Seed-Knoten aus (z. B. dem Cluster-Konfigurationsendpunkt) erkennen. 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 das `dynamicRefreshSources`-Element jedoch auf „wahr“ festgelegt ist, fragt Lettuce alle Knoten ab, um die Cluster-Ansicht 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 `com.sun.jndi.dns.DnsContextFactory`.

Konfigurieren Sie [reconnectDelay](#) mit exponentiellem Backoff und vollständigem Jitter. Lettuce verfügt über integrierte Wiederholungsmechanismen, die auf den exponentiellen Backoff-Strategien basieren. Weitere Einzelheiten finden Sie im Beitrag [Exponentielles Backoff and Jitter](#) im AWS-Architecture-Blog. Weitere Informationen dazu, wie wichtig das Vorhandensein einer Backoff-Strategie für Wiederholungen ist, finden Sie in den Abschnitten zur Backoff-Logik des [Blogbeitrags mit bewährten Methoden](#) im AWS-Datenbank-Blog.

```
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 hinweg iterieren, z. B. FLUSHDB-, FLUSHALL-, KEYS-, SMEMBERS- oder Lua-Skripts. Verwenden Sie kürzere Timeouts für einzelne Schlüsselbefehle wie SET, GET und HSET.

Note

Die im folgenden Beispiel konfigurierten Timeouts gelten für Tests, bei denen SET/GET-Befehle mit Schlüsseln und Werten von bis zu 20 Byte 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 Cluster-Modus und aktiviertem TLS

Note

Die Timeouts im folgenden Beispiel gelten für Tests, bei denen SET/GET-Befehle mit Schlüsseln und Werten von bis zu 20 Byte 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 setJVMPProperties() {
    java.security.Security.setProperty("networkaddress.cache.ttl", "10");
}

private static final Duration META_COMMAND_TIMEOUT = Duration.ofMillis(1000);
private static final Duration DEFAULT_COMMAND_TIMEOUT = Duration.ofMillis(250);
// Socket connect timeout should be lower than command timeout for Lettuce
private static final Duration CONNECT_TIMEOUT = Duration.ofMillis(100);

// Create RedisURI from the cluster configuration endpoint
clusterConfigurationEndpoint = <cluster-configuration-endpoint> // TODO: add your
    cluster configuration endpoint
final RedisURI redisUriCluster =
    RedisURI.Builder.redis(clusterConfigurationEndpoint)
        .withPort(6379)
        .withSsl(true)
        .build();

// Configure the client's resources
ClientResources clientResources = DefaultClientResources.builder()
    .reconnectDelay(
        Delay.fullJitter(
            Duration.ofMillis(100), // minimum 100 millisecond delay
            Duration.ofSeconds(10), // maximum 10 second delay
            100, TimeUnit.MILLISECONDS)) // 100 millisecond base
```

```
.dnsResolver(new DirContextDnsResolver())
    .build();

// Create a cluster client instance with the URI and resources
RedisClusterClient redisClusterClient =
    RedisClusterClient.create(clientResources, redisUriCluster);

// Use a dynamic timeout for commands, to avoid timeouts during
// cluster management and slow operations.
class DynamicClusterTimeout extends TimeoutSource {
    private static final Set<ProtocolKeyword> META_COMMAND_TYPES =
        ImmutableSet.<ProtocolKeyword>builder()
            .add(CommandType.FLUSHDB)
            .add(CommandType.FLUSHALL)
            .add(CommandType.CLUSTER)
            .add(CommandType.INFO)
            .add(CommandType.KEYS)
            .build();

    private final Duration metaCommandTimeout;
    private final Duration defaultCommandTimeout;

    DynamicClusterTimeout(Duration defaultTimeout, Duration metaTimeout)
    {
        defaultCommandTimeout = defaultTimeout;
        metaCommandTimeout = metaTimeout;
    }

    @Override
    public long getTimeout(RedisCommand<?, ?, ?> command) {
        if (META_COMMAND_TYPES.contains(command.getType())) {
            return metaCommandTimeout.toMillis();
        }
        return defaultCommandTimeout.toMillis();
    }
}

TimeoutOptions timeoutOptions = TimeoutOptions.builder()
    .timeoutSource(new DynamicClusterTimeout(DEFAULT_COMMAND_TIMEOUT,
        META_COMMAND_TIMEOUT))
    .build();

// Configure the topology refreshment options
final ClusterTopologyRefreshOptions topologyOptions =
```

```
ClusterTopologyRefreshOptions.builder()
    .enableAllAdaptiveRefreshTriggers()
    .enablePeriodicRefresh()
    .dynamicRefreshSources(true)
    .build();

// Configure the socket options
final SocketOptions socketOptions =
    SocketOptions.builder()
        .connectTimeout(CONNECT_TIMEOUT)
        .keepAlive(true)
        .build();

// Configure the client's options
final ClusterClientOptions clusterClientOptions =
    ClusterClientOptions.builder()
        .topologyRefreshOptions(topologyOptions)
        .socketOptions(socketOptions)
        .autoReconnect(true)
        .timeoutOptions(timeoutOptions)
        .nodeFilter(it ->
            ! (it.is(RedisClusterNode.NodeFlag.FAIL)
                || it.is(RedisClusterNode.NodeFlag.EVENTUAL_FAIL)
                || it.is(RedisClusterNode.NodeFlag.NOADDR)))
        .validateClusterNodeMembership(false)
        .build();

redisClusterClient.setOptions(clusterClientOptions);

// Get a connection
final StatefulRedisClusterConnection<String, String> connection =
    redisClusterClient.connect();

// Get cluster sync/async commands
RedisAdvancedClusterCommands<String, String> sync = connection.sync();
RedisAdvancedClusterAsyncCommands<String, String> async = connection.async();
```

Beispiel: Lettuce-Konfiguration für deaktivierten Cluster-Modus und aktiviertes TLS

Note

Die Timeouts im folgenden Beispiel gelten für Tests, bei denen SET/GET-Befehle mit Schlüsseln und Werten von bis zu 20 Byte 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);
```

Beispiele für IPv6-Clients

Im Folgenden finden Sie bewährte Methoden für die Interaktion mit IPv6-fähigen ElastiCache Ressourcen mithilfe häufig verwendeter Open-Source-Clientbibliotheken. Empfehlungen zur Konfiguration von Clients [für Ressourcen finden Sie in den vorhandenen bewährten Methoden ElastiCache für die Interaktion mit](#). ElastiCache Bei der Interaktion mit IPv6-fähigen Ressourcen gibt es jedoch einige Vorbehalte, die es zu beachten gilt.

Validierte Clients

ElastiCache ist mit Open-Source-Redis kompatibel. Das bedeutet, dass Open-Source-Redis-Clients, die IPv6-Verbindungen unterstützen, in der Lage sein sollten, eine Verbindung zu IPv6 herzustellen,

das für Redis-Cluster aktiviert ist. ElastiCache Darüber hinaus wurden einige der beliebtesten Python- und Java-Clients speziell getestet und validiert, sodass sie mit allen unterstützten Netzwerktypkonfigurationen funktionieren (nur IPv4, nur IPv6 und Dual-Stack)

Validierte Clients:

- [Redis Py \(\) – 4.1.2](#)
- [Lettuce – Version: 6.1.6.RELEASE](#)
- [Jedis – Version: 3.6.0](#)

Konfiguration eines bevorzugten Protokolls für Dual-Stack-Cluster

Bei Redis-Clustern mit aktiviertem Cluster-Modus 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 festgelegt werden.

Für Redis-Cluster legt der IP-Erkennungsparameter das IP-Protokoll fest, das in der Ausgabe von [cluster slots \(\)](#), [cluster shards \(\)](#) und [cluster nodes \(\)](#) verwendet wird. Diese Befehle werden von Clients verwendet, um die Cluster-Topologie zu ermitteln. Clients verwenden die IPs in diesen Befehlen, um eine Verbindung mit 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. Wenn Sie feststellen möchten, wann die Änderungen für einen Redis-Cluster vollständig weitergeleitet wurden, überwachen Sie die Ausgabe von `cluster slots`. Sobald alle vom Befehl „cluster slots“ zurückgegebenen Knoten IPs mit dem neuen Protokoll gemeldet haben, sind die 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.getByAddress(node));
    } catch (UnknownHostException ignored) {}
    return false;
}));
```

TLS-fähige Dual-Stack-Cluster ElastiCache

Wenn TLS für ElastiCache Cluster aktiviert ist, geben die Cluster-Erkennungsfunktionen (`cluster slotscluster shards`, und `cluster nodes`) Hostnamen statt IPs zurück. Die Hostnamen werden dann anstelle von IPs verwendet, um eine Verbindung zum ElastiCache Cluster herzustellen und einen TLS-Handshake durchzuführen. Das bedeutet, dass Clients nicht vom IP-Discovery-Parameter betroffen sind. Bei TLS-fähigen Clustern hat der IP Discovery-Parameter keine

Auswirkung auf das bevorzugte IP-Protokoll. Stattdessen wird das verwendete IP-Protokoll dadurch bestimmt, welches IP-Protokoll der Client bei der Auflösung von DNS-Hostnamen bevorzugt.

Java-Clients

Bei der Verbindung von einer Java-Umgebung aus, die sowohl IPv4 als auch IPv6 unterstützt, bevorzugt Java aus Gründen der Abwärtskompatibilität standardmäßig IPv4 gegenüber IPv6. Die IP-Protokollpräferenz ist jedoch über die JVM-Argumente konfigurierbar. Wenn IPv4 bevorzugt werden soll, akzeptiert die JVM IPv6 `-Djava.net.preferIPv4Stack=true`. Legen Sie `-Djava.net.preferIPv6Stack=true` fest, damit IPv6 bevorzugt wird. Die Einstellung `-Djava.net.preferIPv4Stack=true` bedeutet, dass die JVM keine IPv6-Verbindungen mehr herstellt. Einschließlich der für andere Nicht-Redis-Anwendungen.

Einstellungen auf Host-Ebene

Wenn vom Client oder von der Client-Laufzeit keine Konfigurationsoptionen zum Festlegen einer IP-Protokollpräferenz bereitgestellt wird, hängt das IP-Protokoll bei der DNS-Auflösung im Allgemeinen 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 diejenigen 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 so zu aktualisieren, dass IPv4 bei der Verbindung zu einem ElastiCache Cluster bevorzugt wird, aktualisieren Sie die Priorität für den CIDR-Bereich, der die Cluster-IPs umfasst, so dass sie über der Priorität für Standard-IPv6-Verbindungen liegt. Standardmäßig haben IPv6-Verbindungen eine Priorität von 40. Angenommen, der Cluster befindet sich beispielsweise in einem Subnetz mit dem CIDR `172.31.0.0/16`, die folgende Konfiguration würde dann dazu führen, dass Clients IPv4-Verbindungen mit diesem Cluster vorziehen.

```
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, damit IPv4-Verbindungen gegenüber IPv6-Verbindungen für den angegebenen CIDR-Bereich bevorzugt werden. Gehen wir beispielsweise davon aus, dass sich der Cluster in einem Subnetz mit der CIDR `172.31.0. 0:0 /16` befindet, würde das Ausführen von `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` die folgende Prioritätstabelle ergeben. Das würde dazu führen, dass Clients IPv4 bevorzugen, wenn sie eine Verbindung mit dem Cluster herstellen.

```
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
```

Verwalten von reserviertem Speicher

Reservierter Speicher ist Speicher, der nicht für Daten verwendet werden soll. Bei einer Sicherung oder einem Failover verwendet Redis verfügbaren Speicher zum Aufzeichnen von Schreibvorgängen in Ihrem Cluster, 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 ElastiCache für Redis und zur Anwendung dieser Optionen.

Themen

- [Wie viel reservierten Speicher benötigen Sie?](#)
- [Parameter zum Verwalten von reserviertem Speicher](#)
- [Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers](#)

Wie viel reservierten Speicher benötigen Sie?

Aufgrund der unterschiedlichen ElastiCache Implementierungsmethoden des Sicherungs- und Replikationsprozesses lautet die Faustregel, 25% des `maxmemory` Werts eines Knotentyps mithilfe des `reserved-memory-percent` Parameters zu reservieren. Dies ist der Standardwert und wird in den meisten Fällen empfohlen.

Wenn Burstable-Instance-Typen vom Typ Micro und Small in der Nähe der `maxmemory` Grenzwerte betrieben werden, 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 Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Daten-Tiering](#)

Parameter zum Verwalten von reserviertem Speicher

Seit dem 16. März 2017 bietet Amazon ElastiCache for Redis zwei sich gegenseitig ausschließende Parameter für die Verwaltung Ihres Redis-Speichers, `undreserved-memory` und `reserved-memory-percent`. Keiner dieser Parameter ist Teil der Redis-Distribution.

Je nachdem, wann Sie ElastiCache Kunde wurden, ist der eine oder der andere dieser Parameter der Standardparameter für die Speicherverwaltung. Dieser Parameter gilt, wenn Sie einen neuen Redis-Cluster oder eine Replikationsgruppe erstellen und eine Standardparametergruppe verwenden.

- Für Kunden, die vor dem 16. März 2017 angefangen haben – Wenn Sie einen Redis-Cluster oder eine Replikationsgruppe mit 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 begonnen haben – Wenn Sie einen Redis-Cluster oder eine Replikationsgruppe mit 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 Redis-Speicherverwaltungsparametern vertraut gemacht haben, ziehen Sie es möglicherweise vor, denjenigen zu verwenden, der nicht Ihr Standardwert ist oder dessen Werte nicht standardmäßig eingestellt sind. 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. Diese benutzerdefinierte Parametergruppe können Sie dann beim Erstellen eines Redis-Clusters oder einer Redis-Replikationsgruppe nutzen. 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](#)
- [Erstellen einer Parametergruppe](#)
- [Modifizieren einer Parametergruppe](#)
- [Einen ElastiCache Cluster ändern](#)
- [Ändern einer Replikationsgruppe](#)

Der Parameter "reserved-memory"

Vor dem 16. März 2017 wurde die gesamte ElastiCache für Redis reservierte Speicherverwaltung mit diesem Parameter `reserved-memory` durchgeführt. Der Standardwert von `reserved-memory` ist 0. Mit diesem Standardwert wird kein Speicher für Redis-Overhead reserviert und Redis kann den gesamten Speicher eines Knotens mit Daten belegen.

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 setzen Sie `reserved-memory` auf einen Wert, der für die Redis-Version, die auf Ihrem Cluster ausgeführt wird, und den Knotentyp des Clusters geeignet ist. Weitere Informationen finden Sie unter [Wie viel reservierten Speicher benötigen Sie?](#).

Der Parameter ElastiCache for Redis `reserved-memory` ist spezifisch ElastiCache für Redis und ist nicht Teil der Redis-Distribution.

Die folgenden Schritte zeigen, wie mit `reserved-memory` der Speicher auf Ihrem Redis-Cluster verwaltet wird.

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 [Erstellen einer Parametergruppe](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis6x-m3x1 \  
  --description "Redis 2.8.x for m3.xlarge node type" \  
  --cache-parameter-group-family redis6.x
```

2. Berechnen Sie, wie viele Byte Speicher für Redis-Overhead reserviert werden sollen. Sie finden den Wert von `maxmemory` für Ihren Knotentyp unter [Spezifische Parameter für den Redis-Knotentyp](#).
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 vor 2.8.22 ausführen und die Hälfte der Knoten reservieren müssen. `maxmemory` Weitere Informationen finden Sie unter [Modifizieren einer 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 den Redis-Cluster oder die Redis-Replikationsgruppe so, dass die benutzerdefinierte Parametergruppe verwendet wird.

Mit dem folgenden CLI-Beispiel wird der Cluster `my-redis-cluster` geändert, damit er ab sofort die benutzerdefinierte Parametergruppe `redis28-m3x1` 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
```

Mit dem folgenden CLI-Beispiel wird die Replikationsgruppe `my-redis-repl-grp` geändert, damit sie ab sofort die benutzerdefinierte Parametergruppe `redis28-m3x1` 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
```

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 for Redis 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 ElastiCache for Redis `reserved-memory-percent` ist spezifisch ElastiCache für Redis und nicht Teil der Redis-Distribution.

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](#).

Um Speicher zu reservieren mit `reserved-memory-percent`

Gehen Sie wie folgt vor `reserved-memory-percent`, um den Speicher auf Ihrem ElastiCache for Redis-Cluster zu verwalten:

- Wenn Sie Redis 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 Redis-Version vor 2.8.22 ausführen, müssen Sie wahrscheinlich mehr Speicher als die standardmäßigen 25 Prozent von `reserved-memory-percent` reservieren. 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 [Erstellen einer Parametergruppe](#).

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name redis28-50 \  
  --description "Redis 2.8.x 50% reserved" \  
  --cache-parameter-group-family redis2.8
```

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 [Modifizieren einer 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 Redis-Cluster oder -Replikationsgruppen, auf denen eine Redis-Version vor 2.8.22 ausgeführt wird.

Mit dem folgenden CLI-Beispiel wird der Redis-Cluster `my-redis-cluster` geändert, damit er ab sofort die benutzerdefinierte Parametergruppe `redis28-50` 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 redis28-50 \  
  --apply-immediately
```

Mit dem folgenden CLI-Beispiel wird die Redis-Replikationsgruppe `my-redis-repl-grp` geändert, damit sie ab sofort die benutzerdefinierte Parametergruppe `redis28-50` 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 redis28-50 \  
  --apply-immediately
```

Angabe Ihres Parameters für die Verwaltung reservierten Arbeitsspeichers

Wenn Sie am 16. März 2017 ein aktueller ElastiCache Kunde waren, ist Ihr Standardparameter für die Verwaltung von `reserved-memory` reserviertem Speicher 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 Cluster oder Ihre

Replikationsgruppe ElastiCache für Redis erstellt haben. Sie können jedoch Ihren Parameter für die Verwaltung des reservierten Speichers entweder mithilfe der ElastiCache API AWS CLI oder 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 [Erstellen einer Parametergruppe](#).

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.

Mit dem folgenden CLI-Beispiel wird die benutzerdefinierte Parametergruppe `redis32-cluster-on` geändert, sodass sie `reserved-memory-percent` zum Verwalten von reserviertem Speicher verwendet. 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 [Modifizieren einer 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.

Mit dem folgenden CLI-Beispiel wird die benutzerdefinierte Parametergruppe `redis32-m3x1` geändert, sodass sie `reserved-memory` zum Verwalten von reserviertem Speicher verwendet. 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 [Modifizieren einer 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

Dieser Abschnitt gilt nur, wenn Sie eigene Redis-Cluster entwerfen möchten. 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 Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#)
- [Online-Größenanpassung von Clustern](#)
- [Minimieren der Ausfallzeit während der Wartung](#)

Minimierung von Ausfallzeiten mit Multi-AZ

Weitere Informationen zu [Multi-AZ und ElastiCache zur Minimierung von Ausfallzeiten finden Sie unter Minimierung von Ausfallzeiten in für Redis mit Multi-AZ](#).

Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen

Redis-Snapshots und -Synchronisierungen ab Version 2.8.22

Mit Redis 2.8.22 wird ein unvergabelter Speichervorgang eingeführt, mit dem Sie mehr Arbeitsspeicher zur Verwendung durch Ihre Anwendung zuordnen können, ohne Auslagerungen während Synchronisierungen und Speichervorgängen zu erhöhen. Weitere Informationen finden Sie unter [So werden Synchronisation und Backup implementiert](#).

Redis-Snapshots und -Synchronisierungen vor Version 2.8.22

Wenn Sie mit Redis arbeiten ElastiCache, 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
- Beim Aktivieren der Append-Only-File-Funktion (AOF) für Redis
- Beim Hochstufen eines Replikats zu einem primären (was eine primäre/Replikat-Synchronisierung verursacht).

Bei jedem im Hintergrund ausgeführten Schreibvorgang muss Redis über eine ausreichende Menge an Arbeitsspeicher für den Overhead des Vorgangs verfügen. Wenn nicht genügend Arbeitsspeicher verfügbar ist, schlägt der Vorgang fehl. Daher ist es beim Erstellen Ihres Redis-Clusters wichtig, einen Knoten-Instance-Typ mit ausreichendem Arbeitsspeicher zu wählen.

Hintergrundschriftvorgang und Speicherverwendung

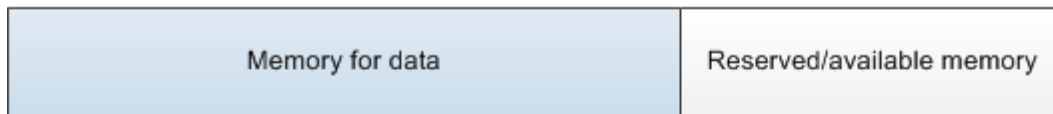
Bei jedem Aufruf eines Hintergrundspeichervorgangs vergabelt Redis seinen Vorgang (zur Erinnerung, Redis verwendet Single-Thread-Vorgänge). Eine Vergabelung erhält die Daten dauerhaft in einer Redis .rdb snapshot-Datei auf der Festplatte. Die andere Vergabelung führt alle Lese- und Schreibvorgänge durch. Um sicherzustellen, dass es sich bei Ihrem point-in-time Snapshot um einen 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:

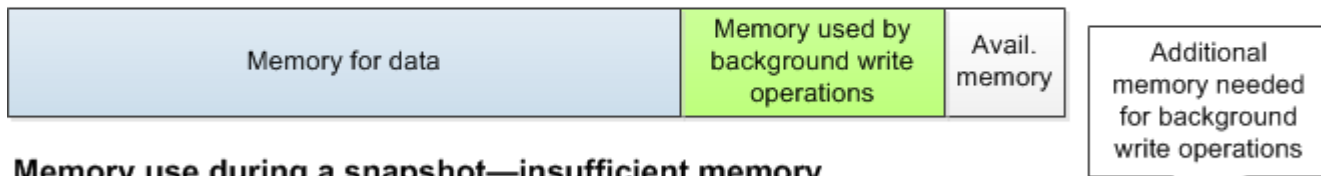
- Ihre Anwendung führt viele Schreibvorgänge aus und benötigt daher eine große Menge an verfügbarem Arbeitsspeicher zum Akzeptieren neuer oder aktualisierter Daten.
- Es ist sehr wenig Arbeitsspeicher zum Schreiben neuer oder aktualisierter Daten verfügbar.
- Die dauerhafte Erhaltung eines großen Datensatzes 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 Redis Snapshots ausführt, finden Sie unter <http://redis.io>.

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Availability Zones](#).

Vermeidung von Speichermangel beim Ausführen eines Hintergrundschreibvorgangs

Bei jedem Aufruf eines Hintergrundschreibvorgangs wie BGSAVE oder BGREWRITEAOF muss mehr Arbeitsspeicher verfügbar sein, als von den Schreibvorgängen bei diesem Vorgang verbraucht wird, damit der Vorgang erfolgreich ist. Als Worst-Case-Szenario werden während des Hintergrundschreibvorgangs alle einzelnen Redis-Datensätze aktualisiert und einige neue Datensätze zum Cache hinzugefügt. Aus diesem Grund empfehlen wir, dass Sie `reserved-memory-percent` für Redis-Versionen vor 2.8.22 auf 50 (50 Prozent) oder für Redis-Versionen 2.8.22 und höher auf 25 (25 Prozent) festlegen.

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“. Dies erlaubt Redis, die gesamte Menge von `maxmemory` für Daten zu nutzen, wodurch potenziell zu wenig Arbeitsspeicher für andere Verwendungszwecke wie Hintergrundschreibvorgänge übrig bleibt. Informationen zu `maxmemory`-Werten nach Knoten-Instance-Typ finden Sie unter [Spezifische Parameter für den Redis-Knotentyp](#).

Sie können mit dem Parameter `reserved-memory` auch die Speichermenge verringern, die Redis für das Feld verwendet.

Weitere Informationen zu Redis-spezifischen Parametern finden Sie unter ElastiCache. [Redis-spezifische Parameter](#)

Weitere Informationen zum Erstellen und Ändern von Parametergruppen finden Sie unter [Erstellen einer Parametergruppe](#) und [Modifizieren einer 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. Wir empfehlen daher, die CPU-Auslastung beim Resharding auf Mehrkern-Instances unter 80 Prozent und auf Einzelkern-Instances unter 50 Prozent zu halten. Überwachen Sie ElastiCache-for-Redis-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.
- Ihre Funktionen auf jeden Shard laden – bei der Aufskalierung Ihres Clusters repliziert ElastiCache automatisch die Funktionen, die in einem der (zufällig ausgewählten) vorhandenen Knoten geladen sind, auf den bzw. die neuen Knoten. Wenn Ihr Cluster über Redis 7.0 oder höher verfügt und Ihre Anwendung [Redis Functions](#) (Redis-Funktionen) verwendet, empfehlen wir, alle Ihre Funktionen vor der Aufskalierung auf alle Shards zu laden, damit Ihr Cluster am Ende keine unterschiedlichen Funktionen auf verschiedenen Shards aufweist.

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 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 in dem Cluster AutoFailover aktiviert 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. Wenn AutoFailover deaktiviert ist, ändert sich die Rolle des Knotens nicht. Die Ausfallzeit bei verwalteten oder nicht verwalteten Operationen ist jedoch länger als bei Clustern, bei denen AutoFailover aktiviert ist.

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.

Bewährte Methoden für Redis

Im Folgenden finden Sie bewährte Methoden für die Verwendung von Redis zur Verbesserung der Leistung und Zuverlässigkeit:

- Konfigurationen mit aktiviertem Cluster-Modus verwenden – Mit aktiviertem Cluster-Modus kann der Cache horizontal skaliert werden, um mehr Speicher und Durchsatz zu erzielen als bei einer Konfiguration mit deaktiviertem Cluster-Modus. ElastiCache Serverless ist nur in einer Konfiguration mit aktiviertem Cluster-Modus verfügbar.
- Langlebige Verbindungen verwenden – 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 Serverless 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.

In ElastiCache Serverless leitet das Lesen vom Replikat-Port (6380) die Lesevorgänge nach Möglichkeit an die lokale Availability Zone des Clients weiter, 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 Redis zur Unterstützung von Pub/Sub-Workloads mit hohem Durchsatz nutzen, empfehlen wir die Verwendung von [Sharded Pub/Sub](#) (verfügbar mit Redis 7 oder höher). Herkömmliche Pub/Sub-Cluster mit aktiviertem Cluster-Modus senden Nachrichten an alle Knoten im Cluster, was zu hoher EngineCPUUtilization führen kann. Beachten Sie, dass herkömmliche Pub/Sub-Befehle in ElastiCache Serverless intern Sharded Pub/Sub-Befehle verwenden.

Caching-Strategien

Im folgenden Thema finden Sie Strategien zum Füllen und Verwalten Ihres 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 von 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 In-Memory-Schlüsselwertspeicher, der sich zwischen Ihrer Anwendung und dem Datenspeicher (Datenbank) befindet, auf den sie zugreift. Immer wenn Ihre Anwendung Daten anfordert, stellt sie die Anfrage zuerst an den ElastiCache-Cache. Wenn die Daten im Cache vorhanden und aktuell sind, gibt ElastiCache die Daten an Ihre Anwendung zurück. 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 von 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-Werts \(TTL\)](#) 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 von 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 Time-to-Live-Wert (TTL) hinzufügen, können Sie die Vorteile jeder Strategie 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 bis zum Ablauf des Schlüssels angibt. Redis kann Sekunden oder Millisekunden für diesen Wert angeben. 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 dem [Redis-Befehl set](#) oder dem .

Beispiele für TTL Pseudocode

Das Folgende ist 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 für Redis](#)

Verwalten Ihrer selbst entworfenen Cluster

Dieser Abschnitt enthält Themen, die Ihnen bei der Verwaltung Ihrer selbst entworfenen Cluster helfen.

Note

Diese Themen gelten nicht für ElastiCache Serverless.

Themen

- [Auto Scaling ElastiCache für Redis-Cluster](#)
- [Ändern des Clustermodus](#)
- [AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher](#)
- [Hohe Verfügbarkeit mit Replikationsgruppen](#)
- [Verwaltung der Wartung](#)
- [Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen](#)

Auto Scaling ElastiCache für Redis-Cluster

Voraussetzungen

ElastiCache für Redis ist Auto Scaling auf Folgendes beschränkt:

- Redis-Cluster (Cluster-Modus aktiviert) mit Redis-Engine ab Version 6.0
- Daten-Tiering-Cluster (Cluster-Modus aktiviert) mit Redis-Engine ab Version 7.0.7
- Instance-Größen - Large, XLarge, 2xLarge
- Instance-Familien – R7g, R6g, R6gd, R5, M7g, M6g, M5, C7gn
- Auto Scaling in ElastiCache for Redis wird nicht für Cluster unterstützt, die in globalen Datenspeichern, Outposts oder Local Zones ausgeführt werden.

Automatisches Kapazitätsmanagement mit ElastiCache for Redis Auto Scaling

ElastiCache Für Redis ist Auto Scaling die Fähigkeit, die gewünschten Shards oder Replikate in Ihrem ElastiCache for Redis-Service automatisch zu erhöhen oder zu verringern. ElastiCache for Redis 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 für Redis verwendet Auto Scaling die

Richtlinie, um die Anzahl der Instanzen als Reaktion auf tatsächliche Workloads zu erhöhen oder zu verringern.

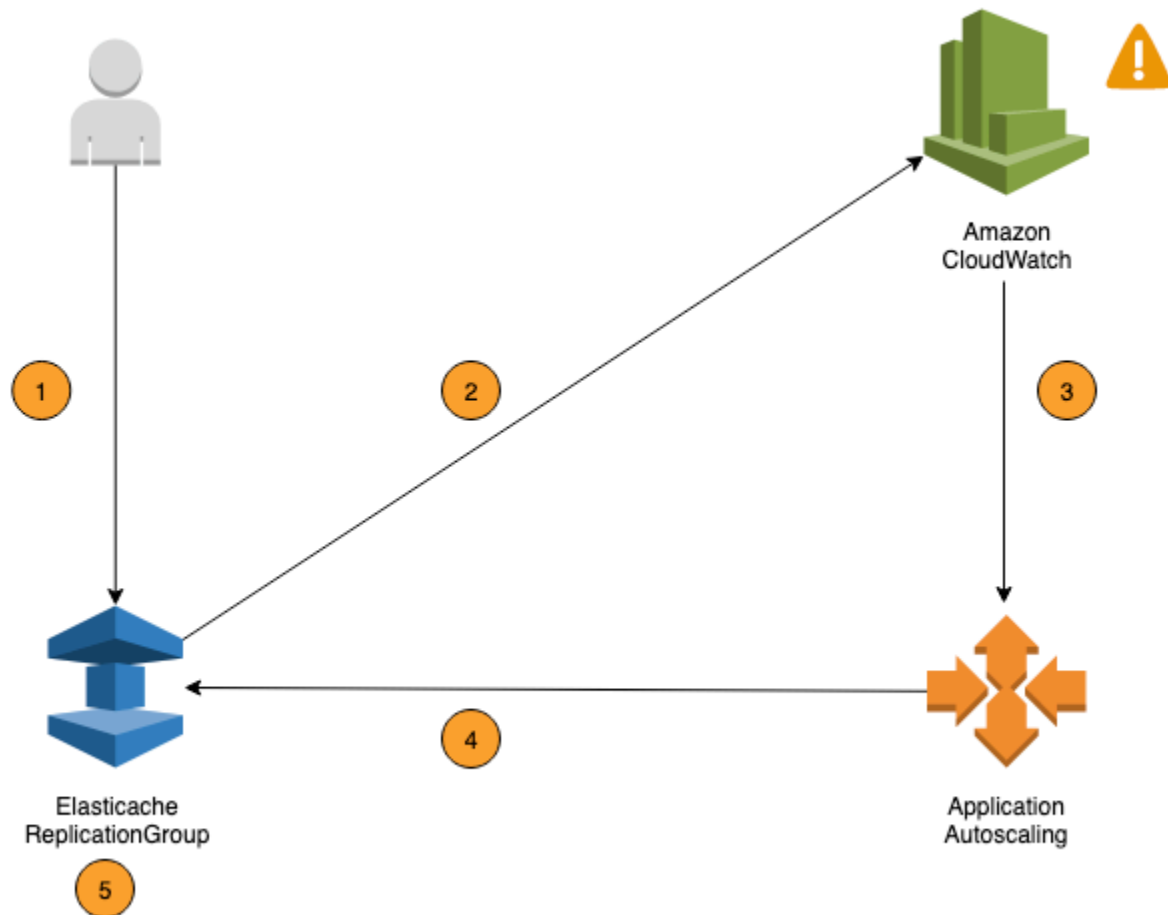
Sie können die verwenden AWS Management Console , um eine Skalierungsrichtlinie anzuwenden, die auf einer vordefinierten Metrik basiert. Eine `predefined metric` ist in einer Aufzählung definiert, sodass Sie sie im Code durch einen Namen angeben oder in der AWS Management Console verwenden können. Benutzerdefinierte Metriken können nicht über die AWS Management Console ausgewählt werden. Alternativ können Sie entweder die AWS CLI oder die Application Auto Scaling-API verwenden, um eine Skalierungsrichtlinie anzuwenden, die auf einer vordefinierten oder benutzerdefinierten Metrik basiert.

ElastiCache for Redis unterstützt die Skalierung für die folgenden Dimensionen:

- **Shards**— Automatisches Hinzufügen/Entfernen von Shards im Cluster ähnlich wie beim manuellen Online-Resharding. In diesem Fall löst Auto Scaling ElastiCache für Redis die Skalierung in Ihrem Namen aus.
- **Replicas**— Automatisches Hinzufügen/Entfernen von Replikaten im Cluster ähnlich wie bei manuellen Erhöhen/Verringern von Replikaten. ElastiCache für Redis fügt Auto Scaling Replikate einheitlich für alle Shards im Cluster hinzu oder entfernt sie.

ElastiCache for Redis unterstützt die folgenden Arten von Richtlinien für die automatische Skalierung:

- [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 Application ElastiCache for Redis 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 auto Skalierungsprozess ElastiCache für Redis zusammen, wie im vorherigen Diagramm dargestellt:

1. Sie erstellen eine Auto Scaling-Richtlinie ElastiCache für Redis ElastiCache für Ihre Redis-Replikationsgruppe.
2. ElastiCache für Redis erstellt Auto Scaling 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 Metrikerwert Ihre Zielauslastung für einen bestimmten Zeitraum überschreitet (oder unter das Ziel fällt), wird ein Alarm CloudWatch ausgelöst, der Redis Auto Scaling ElastiCache zur Bewertung Ihrer Skalierungsrichtlinie auslöst.
4. ElastiCache für Redis gibt Auto Scaling eine Modifizierungsanforderung aus, um Ihre Clusterkapazität anzupassen.

5. ElastiCache für Redis verarbeitet die Modifizierungsanforderung und erhöht (oder verringert) die Kapazität der Cluster-Shards/Replicas dynamisch, sodass sie sich Ihrer Zielauslastung annähert.

Um zu verstehen, wie Auto Scaling ElastiCache für Redis funktioniert, nehmen wir an, Sie haben einen Cluster mit dem Namen `UsersCluster`. Durch die Überwachung der CloudWatch Metriken bestimmen Sie die maximale Anzahl an `ShardsUsersCluster`, die der Cluster benötigt, wenn der Verkehr seinen Höhepunkt erreicht, und die Mindestanzahl an Shards, wenn der Verkehr am niedrigsten Punkt ist. Sie bestimmen auch einen Zielwert für die CPU-Auslastung für die `UsersCluster`-Cluster erstellt. ElastiCache für Redis verwendet Auto Scaling 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 für Redis ändert Auto Scaling die Ressourceneinstellungen nur, wenn die tatsächliche Arbeitslast über einen längeren Zeitraum von mehreren Minuten erhöht (oder reduziert) bleibt. Der Target-Tracking-Algorithmus ElastiCache für Redis Auto Scaling 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 for Redis Auto Scaling verwendet, um zu bestimmen, wann und wieviel zu skalieren ist.
- Minimale und maximale Kapazität — Die minimale und die maximale Anzahl an Shards oder Replikaten, die für die Skalierung verwendet werden sollen.

Important

Wenn die aktuelle Kapazität bei der Erstellung der automatischen Skalierungsrichtlinie höher ist als die konfigurierte maximale Kapazität, skalieren wir während der Richtlinienerstellung auf die `MaxCapacity`. Ähnlich, wenn die aktuelle Kapazität niedriger ist als `min Kapazität` konfiguriert, wir `ScaleOut` auf die `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 service-verknüpfte Rolle —AWS Die Identity and Access Management (IAM) - Rolle ist mit einer bestimmten AWS-Service eröffnen. Eine serviceverknüpfte Rolle umfasst alle Berechtigungen, die der Service für den Aufruf anderer AWS-Services in Ihrem Namen benötigt. ElastiCache for Redis Auto Scaling generiert automatisch diese Rolle, `AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG`, für Sie.
- 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 Richtlinien wird eine vordefinierte oder benutzerdefinierte Metrik und ein Zielwert für die Metrik in einer Konfiguration der Skalierungsrichtlinien für die Zielverfolgung festgelegt. ElastiCache for Redis Auto Scaling erstellt und verwaltet CloudWatch-Alarme, die die Skalierungsrichtlinie auslösen und berechnet die Skalierungsanpassung basierend auf der Metrik und dem Zielwert. 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. Eine solche Richtlinie kann die CPU-Auslastung bei einem bestimmten Prozentsatz der Auslastung halten, z. B. bei 70 Prozent.

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 an Shards angeben, die von Auto Scaling von ElastiCache für Redis skaliert werden können. Dieser Wert muss kleiner oder gleich 250 sein und mindestens 1 sein. Sie können auch die minimale Anzahl an Shards angeben, die von Auto Scaling von ElastiCache für Redis 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 der Replikate angeben, die von ElastiCache for Redis Auto Scaling verwaltet werden sollen. Dieser Wert muss gleich oder kleiner 5 sein. Sie können die minimale Anzahl an Replikaten angeben, auf die von ElastiCache for Redis Auto Scaling verwaltet werden kann. 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 for Redis Auto-Scaling-Richtlinien erhöhen die Cluster-Kapazität, bis sie Ihre definierte maximale Größe erreicht oder bis Service-Limits gelten. Um eine Erhöhung des Limits zu beantragen, [AWSsiehe 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 Datenverkehr einer Variante gleich Null wird, skaliert ElastiCache for Redis automatisch auf die Mindestanzahl der angegebenen Instances herunter.

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. Dadurch werden die Löschungen von Shards und -Replicas in Ihrem ElastiCache for Redis-Cluster für Scale-In-Anforderungen sowie die Erstellung von Shards bzw. Replicas für Scale-Out-Anfragen verlangsamt. Sie können die folgenden Ruhephasen angeben:

- Eine Scale-in-Aktivität reduziert die Anzahl der Shards / Replikate in Ihrem ElastiCache-for-Redis-Cluster. Eine Scale-In-Ruhephase gibt die Zeitspanne in Sekunden an, nach der eine Scale-In-Aktivität abgeschlossen sein muss, bevor eine weitere Scale-In-Aktivität gestartet werden kann.
- Eine Scale-out-Aktivität erhöht die Anzahl der Shards / Replikate in Ihrem ElastiCache-for-Redis-Cluster. Eine Scale-Out-Ruhephase gibt die Zeitspanne in Sekunden an, nach der eine Scale-Out-Aktivität abgeschlossen sein muss, bevor eine weitere Scale-Out-Aktivität 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.

ℹ Note

Scale-out-Aktivitäten sind immer aktiviert, damit die Skalierungsrichtlinie ElastiCache-for-Redis-Shards / Replikate nach Bedarf erstellen kann.

Für ElastiCache Redis Auto Scaling sind IAM-Berechtigungen erforderlich

ElastiCache für Redis Auto Scaling wird durch eine Kombination der APIs ElastiCache for Redis und Application Auto Scaling ermöglicht. CloudWatch Cluster werden mit ElastiCache für Redis erstellt und aktualisiert, Alarme werden mit CloudWatch Application Auto Scaling erstellt und Skalierungsrichtlinien werden erstellt. Zusätzlich zu den Standard-IAM-Berechtigungen für das Erstellen und Aktualisieren von Clustern muss der IAM-Benutzer, der auf die Redis Auto Scaling Scaling-Einstellungen zugreift ElastiCache , über die entsprechenden Berechtigungen für die Dienste verfügen, die dynamische Skalierung unterstützen. IAM-Benutzer müssen die Berechtigung haben, die Aktionen in der folgenden Beispielrichtlinie zu verwenden:

```
{
  "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 Auto Scaling-Dienst ElastiCache für Redis benötigt außerdem die Erlaubnis, Ihre Cluster und CloudWatch Alarme zu beschreiben, sowie Berechtigungen, Ihre Zielkapazität ElastiCache für Redis in Ihrem Namen zu ändern. Wenn Sie Auto Scaling für Ihren ElastiCache for Redis-Cluster aktivieren, wird eine dienstverknüpfte Rolle mit dem Namen erstellt.

`AWSServiceRoleForApplicationAutoScaling_ElastiCacheRG` Diese dienstbezogene Rolle gewährt ElastiCache Redis Auto Scaling die Erlaubnis, 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 ElastiCache für Redis Auto Scaling. Weitere Informationen finden Sie unter [Dienstbezogene Rollen ElastiCache für Redis 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 – Ermitteln Sie, ob Ihr Cluster über CPU- oder datenintensive Workloads verfügt, und verwenden Sie eine entsprechende vordefinierte Metrik, um die Skalierungsrichtlinie zu definieren.
 - Engine-CPU: `ElastiCachePrimaryEngineCPUUtilization` (Shard-Dimension) oder `ElastiCacheReplicaEngineCPUUtilization` (Replikatdimension)
 - Datenbanknutzung:
`ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` Diese Skalierungsrichtlinie funktioniert am besten, wenn „maxmemory-policy“ im Cluster auf „noeviction“ festgelegt ist.

Wir empfehlen Ihnen, mehrere Richtlinien pro Dimension auf dem Cluster zu vermeiden.

ElastiCache Für Redis skaliert Auto Scaling das skalierbare Ziel, wenn irgendwelche Ziel-Tracking-Richtlinien für die Skalierung bereit sind, aber es wird nur dann skaliert, wenn alle Ziel-Tracking-Richtlinien (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 Redis 7.0 oder höher verfügt und Ihre Anwendung [Redis Functions](#) (Redis-Funktionen) verwendet, empfehlen wir, alle Ihre Funktionen vor der Aufskalierung auf alle Shards zu laden, damit Ihr Cluster am Ende keine unterschiedlichen Funktionen auf verschiedenen Shards aufweist.

Beachten Sie nach der Registrierung für 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

Im Folgenden finden Sie Details zur Zielverfolgung und geplanten Richtlinien und zum Anwenden dieser Richtlinien mithilfe der 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 for Redis Auto Scaling erstellt und verwaltet die CloudWatch-Alarme, die die Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung basierend auf der Metrik und dem Zielwert. 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. Eine solche Richtlinie kann die CPU-Nutzung auf oder nahe an dem Zielwert halten.

Vordefinierte Metriken

Eine vordefinierte Metrik ist eine Struktur, die sich auf einen bestimmten Namen, eine bestimmte 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-Metrikenname | CloudWatch-Metrikdimension | Nicht geeignete Instance-Typen |
|--|---|---|--------------------------------|
| <code>ElastiCachePrimaryEngineCPUUtilization</code> | <code>EngineCPUUtilization</code> | ReplicationGroupId, Rolle = Primär | Keine |
| <code>ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage</code> | <code>DatabaseCapacityUsageCountedForEvictPercentage</code> | Metriken für die Redis-Replikationsgruppe | Keine |
| <code>ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage</code> | <code>DatabaseMemoryUsageCountedForEvictPercentage</code> | Metriken für die Redis-Replikationsgruppe | R6gd |

Instance-Typen mit Daten-Tiering können

`ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage` nicht verwenden, da diese Instance-Typen Daten sowohl im Arbeitsspeicher als auch im SSD-Speicher ablegen. Der erwartete Anwendungsfall für Instances mit Daten-Tiering ist eine 100-prozentige Speichernutzung und das Auffüllen des SSD-Speichers nach Bedarf.

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 für Redis skaliert Ihre Cluster-Shards um eine Anzahl, die der größeren von zwei Zahlen entspricht: prozentuale Variation von Target und 20 Prozent der aktuellen Shards. Beim Scale-In kann ElastiCache for Redis nicht automatisch skaliert werden, es sei denn, der Gesamtwert liegt unter 75 Prozent des definierten Ziels.

Wenn Sie für ein Scal-Out-Beispiel 50 Shards und

- Wenn Ihr Target um 30 Prozent verletzt, skaliert ElastiCache for Redis um 30 Prozent, was zu 65 Shards pro Cluster führt.
- Wenn Ihr Target um 10 Prozent verletzt, skaliert ElastiCache for Redis standardmäßig Minimum von 20 Prozent, was zu 60 Shards pro Cluster führt.

Wenn Sie für ein Scale-In-Beispiel einen Target-Wert von 60 Prozent ausgewählt haben, wird ElastiCache for Redis nicht automatisch skalieren, bis die Metrik kleiner oder gleich 45 Prozent ist (25 Prozent unter dem Ziel 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 Ziel-Nachverfolgung verwenden, um eine horizontale Skalierung nach oben vorzunehmen, wenn die angegebene Metrik unter dem Zielwert liegt. ElastiCache für Redis skaliert Shards um mindestens 20 Prozent Abweichung des Ziels vorhandener Shards im Cluster.
- 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. Der Grund hierfür ist, dass ElastiCache for Redis Auto Scaling stets vorsichtig agiert, indem beim Ermitteln der hinzuzufügenden oder zu entfernenden Kapazität Auf- oder Abrundungen vorgenommen werden. 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 Zielverfolgung-Skalierungsrichtlinien für einen ElastiCache-for-Redis-Cluster haben, vorausgesetzt, dass jede von ihnen eine andere Metrik verwendet. Die Absicht von ElastiCache for Redis Auto Scaling ist es, immer die Verfügbarkeit zu priorisieren, 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 for Redis Auto Scaling für eine Skalierungsrichtlinie für die Zielverfolgung verwaltet. ElastiCache for Redis Auto Scaling löscht die Alarme automatisch, wenn Sie die Skalierungsrichtlinie löschen.
- ElastiCache for Redis Auto Scaling hindert Sie nicht daran, Cluster-Shards manuell zu ändern. Diese manuellen Anpassungen haben keinerlei Auswirkungen auf bestehende CloudWatch Alarme, die an die Skalierungsrichtlinie angehängt sind, aber sie können sich auf Metriken auswirken, die diese CloudWatch-Alarme auslösen können.
- Diese CloudWatch Alarme, die von Auto Scaling verwaltet werden, werden über die AVG-Metrik für alle Shards im Cluster definiert. So kann Hot-Shards zu einem beliebigen Szenario führen:
 - Skalierung, wenn nicht erforderlich, da einige Hot Shards geladen werden, die einen CloudWatch Alarm auslösen
 - nicht skalieren, wenn dies erforderlich ist, aufgrund aggregierter AVG über alle Shards, die den Alarm nicht verletzen.
- ElastiCache for Redis-Standardlimits für Knoten pro Cluster gelten weiterhin. 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 Ihrer VPC genügend ENIs (Elastic Network Interfaces) zur Verfügung stehen, die während der Skalierung erforderlich sind. Weitere Informationen finden Sie unter [Elastic-Network-Schnittstellen](#).

- Wenn nicht genügend Kapazität von EC2 zur Verfügung steht, würde ElastiCache for Redis Auto Scaling nicht skalieren und verzögert, bis die Kapazität verfügbar ist.
- ElastiCache für Redis Auto Scaling während des Scale-Ins entfernt keine Shards mit Slots mit einer Elementgröße größer als 256 MB nach der Serialisierung.
- 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 über die bearbeitete AWS Management Console.

So fügen Sie eine Auto-Scaling-Richtlinie zu einem ElastiCache-for-Redis-Cluster hinzu

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich 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. 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-Auslastung, um eine Richtlinie auf der Grundlage der durchschnittlichen CPU-Auslastung zu erstellen.
 - 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 die Speicher- und SSD-Auslastung für Daten-Tiering-Instances und die Speichernutzung 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 in Ihren ElastiCache-Shards bei:

- Primäre CPU-Auslastung: behält den Zielwert für die `EngineCPUUtilization`-Metrik auf den Primärknoten 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 der AWS CLI, um die Abklingzeitwerte zu ändern.
11. Geben Sie für Mindestkapazität die Mindestanzahl von Shards ein, die die Richtlinie von ElastiCache for Redis Auto Scaling aufrechterhalten muss.
12. Geben Sie für Maximalkapazität die maximale Anzahl von Shards ein, die die Richtlinie von ElastiCache for Redis Auto Scaling aufrechterhalten muss. Dieser Wert muss gleich oder kleiner 250 sein.
13. Wählen Sie Erstellen aus.

Registrieren eines skalierbaren Ziels

Bevor Sie Auto Scaling mit einem ElastiCache for Redis-Cluster verwenden können, müssen Sie zunächst Ihren Cluster bei ElastiCache for Redis Auto Scaling registrieren. Damit legen Sie die Skalierungsdimension und die Grenzen fest, die auf diesen Cluster angewendet werden sollen. ElastiCache für Redis automatische Skalierung skaliert den ElastiCache for Redis-Cluster dynamisch entlang `derelasticache:replication-group:NodeGroupsskalierbare` Dimension, die die Anzahl der Cluster-Shards darstellt.

Verwendung der AWS CLI

Um Ihren ElastiCache for Redis-Cluster zu registrieren, verwenden Sie die [register-scalable-target-](#)Befehl mit den folgenden Parametern:

- `--service-namespace` – Legen Sie diesen Wert auf `elasticache` fest
- `--resource-id`— Die Ressourcenkennung für den ElastiCache for Redis-Cluster. Für diesen Parameter lautet der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung

ist der Name des ElastiCache-for-Redis-Clusters, beispielsweise `replication-group/myscalablecluster`.

- `--scalable-dimension` – Legen Sie diesen Wert auf `elasticache:replication-group:NodeGroups` fest.
- `--max-capacity` — Die maximale Anzahl an Shards, die von ElastiCache for Redis Auto Scaling von Redis verwaltet werden sollen. 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 minimale Anzahl an Shards, die von ElastiCache for Redis Auto Scaling von Redis verwaltet werden sollen. 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 einen ElastiCache-for-Redis-Cluster 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 ^  

```

Verwenden der API

Wenn Sie Ihren ElastiCache-Cluster registrieren möchten, verwenden Sie den Befehl [register-scalable-target](#) mit den folgenden Parametern:

- `ServiceNamespace` — Legen Sie diesen Wert auf `elasticache`.
- `ResourceId` – die Ressourcenkennung für den ElastiCache-Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner ist der Name des ElastiCache-for-Redis-Clusters, zum Beispiel `replication-group/myscalablecluster`.
- `ScalableDimension` – Legen Sie diesen Wert auf `elasticache:replication-group:NodeGroups` fest.
- `MinCapacity` — Die minimale Anzahl an Shards, die von Auto Scaling von ElastiCache für Redis verwaltet werden sollen. 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 an Shards, die von Auto Scaling von ElastiCache für Redis verwaltet werden sollen. 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-for-Redis-Cluster mit dem Namen `myscalablecluster` mit der 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/myscalablecluster",
  "ScalableDimension": "elasticache:replication-group:NodeGroups",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definieren einer Skalierungsrichtlinie

Eine Konfiguration der Skalierungsrichtlinie für die Zielverfolgung wird durch einen JSON-Block repräsentiert, in dem die Metriken und Zielwerte definiert sind. Sie können die Konfiguration einer Skalierungsrichtlinie als JSON-Block in einer Textdatei speichern. Sie verwenden diese Textdatei, wenn Sie die AWS CLI oder die Application Auto Scaling Scaling-API aufrufen. Weitere Informationen zur Syntax der Richtlinienkonfiguration finden Sie [TargetTrackingScalingPolicyConfiguration](#) in der Referenz zur Application Auto Scaling API.

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](#)
- [Anwenden einer Skalierungsrichtlinie](#)

Verwenden einer vordefinierten Metrik

Mithilfe vordefinierter Metriken können Sie schnell eine Skalierungsrichtlinie für die Zielverfolgung für einen ElastiCache für Redis Cluster definieren, die mit der Zielverfolgung in ElastiCache Redis Auto Scaling funktioniert.

Derzeit unterstützt ElastiCache for Redis die folgenden vordefinierten Metriken in ElastiCache Redis NodeGroup Auto Scaling:

- `ElastiCachePrimaryEngineCPUUtilization` — Der Durchschnittswert der `EngineCPUUtilization` Metrik für alle primären Knoten CloudWatch im for Redis-Cluster. ElastiCache
- `ElastiCacheDatabaseMemoryUsageCountedForEvictPercentage`— Der Durchschnittswert der `DatabaseMemoryUsageCountedForEvictPercentage` Metrik CloudWatch auf allen primären Knoten im ElastiCache for Redis-Cluster.
- `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage`— Der Durchschnittswert der `ElastiCacheDatabaseCapacityUsageCountedForEvictPercentage` Metrik CloudWatch auf allen Primärknoten im ElastiCache for Redis-Cluster.

Weitere Informationen über die Metriken

EngineCPUUtilization, DatabaseMemoryUsageCountedForEvictPercentage und DatabaseCapacity. Sie unter [Überwachung der Nutzung mithilfe von CloudWatch-Metriken](#). Um eine vordefinierte Metrik in Ihrer Skalierungsrichtlinie zu verwenden, erstellen Sie eine Zielverfolgungskonfiguration für Ihre Skalierungsrichtlinie. Diese Konfiguration muss a PredefinedMetricSpecification für die vordefinierte Metrik und a TargetValue für den Zielwert dieser Metrik enthalten.

Example

Das folgende Beispiel beschreibt eine typische Richtlinienkonfiguration für die Skalierung nach Zielverfolgung ElastiCache für einen For-Redis-Cluster. In dieser Konfiguration wird die ElastiCachePrimaryEngineCPUUtilization vordefinierte Metrik verwendet, um den ElastiCache For-Redis-Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent auf allen 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 definieren, die auf einer beliebigen ElastiCache Metrik basiert, 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 für Redis-Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 50 Prozent für alle Shards in einem genannten Cluster 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 für 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 For-Redis-Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent auf allen Primärknoten 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": "ElastiCachePrimaryEngineCPUUtilization"
  },
  "ScaleInCooldown": 600,
```

```
"ScaleOutCooldown": 300
}
```

Deaktivieren der Scale-In-Aktivität

Sie können verhindern, dass die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung in Ihrem ElastiCache for Redis-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 für Application Auto Scaling.

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt die `ElastiCachePrimaryEngineCPUUtilization` vordefinierte Metrik einen ElastiCache für Redis-Cluster auf der Grundlage einer durchschnittlichen CPU-Auslastung von 40 Prozent für alle 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 bei ElastiCache for Redis Auto Scaling registriert und eine Skalierungsrichtlinie definiert haben, wenden Sie die Skalierungsrichtlinie auf den registrierten Cluster an. Um eine Skalierungsrichtlinie auf einen ElastiCache for Redis-Cluster anzuwenden, können Sie die AWS CLI oder die Application Auto Scaling Scaling-API verwenden.

Anwenden einer Skalierungsrichtlinie mithilfe der AWS CLI

Um eine Skalierungsrichtlinie auf Ihren ElastiCache for Redis-Cluster 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 die für Redis. ElastiCache Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des ElastiCache for Redis-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 Redis-Cluster verwendet werden soll. ElastiCache

Im folgenden Beispiel wenden Sie eine Skalierungsrichtlinie `myscalablepolicy` zur Zielverfolgung an, die auf einen ElastiCache for Redis-Cluster benannt ist, der `myscalablecluster` mit ElastiCache for 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: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
```

Anwendung einer Skalierungsrichtlinie mit der API

Um eine Skalierungsrichtlinie auf Ihren ElastiCache for Redis-Cluster anzuwenden, verwenden Sie den [PutScalingPolicy](#) AWS CLI Befehl mit den folgenden Parametern:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID für die für Redis. ElastiCache Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner beispielsweise der Name des ElastiCache for Redis-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 Redis-Cluster verwendet werden soll. ElastiCache

Im folgenden Beispiel wenden Sie eine Skalierungsrichtlinie `myscalablepolicy` zur Zielverfolgung an, die auf einen ElastiCache for Redis-Cluster benannt ist, der `myscalablecluster` mit ElastiCache for Redis Auto Scaling benannt ist. 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 über die AWS Management Console, die AWS CLI oder die Application-Auto-Scaling-API bearbeiten.

Bearbeiten einer Skalierungsrichtlinie über die AWS Management Console

So bearbeiten Sie eine Auto-Scaling-Richtlinie für einen ElastiCache-for-Redis-Cluster

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich 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 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 über das AWS CLI und die API

Über die AWS CLI oder die Application-Auto-Scaling-API können Sie eine Skalierungsrichtlinie auf die gleiche Weise bearbeiten, wie Sie sie anwenden:

- Wenn Sie die AWS CLI verwenden, geben Sie den Namen der zu bearbeitenden Richtlinie im Parameter `--policy-name` an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.
- Wenn Sie die Application Auto Scaling API verwenden, geben Sie den Namen der zu bearbeitenden Richtlinie im Parameter `PolicyName` 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 über die AWS Management Console, die AWS CLI oder die Application-Auto-Scaling-API löschen.

Löschen einer Skalierungsrichtlinie über die AWS Management Console

So löschen Sie eine Auto-Scaling-Richtlinie für einen ElastiCache-for-Redis-Cluster

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis 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 über die AWS CLI

Um eine Skalierungsrichtlinie für Ihren ElastiCache for Redis-Cluster zu löschen, verwenden Sie die [Löschen-Skalierung-Richtlinie](#) AWS CLI-Befehl mit den folgenden Parametern:

- `—policy-name—` Der Name der Skalierungsrichtlinie.
- `—resource-id—` Der Ressourcen-Bezeichner für ElastiCache for Redis. Für diesen Parameter lautet der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung ist der Name des ElastiCache-for-Redis-Clusters, beispielsweise `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 Zielverfolgungs-Skalierungsrichtlinie namens `myscalablepolicy` aus einem ElastiCache-for-Redis-Cluster namens `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 über die API

Um eine Skalierungsrichtlinie aus Ihrem [ElastiCache-for-Redis-Cluster zu löschen, verwenden Sie den AWS CLI-Befehl DeleteScalingPolicy](#) mit den folgenden Parametern:

- `—policy-name—` Der Name der Skalierungsrichtlinie.
- `—resource-id—` Der Ressourcen-Bezeichner für ElastiCache for Redis. Für diesen Parameter lautet der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung ist der Name des ElastiCache-for-Redis-Clusters, beispielsweise `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 Zielverfolgungs-Skalierungsrichtlinie namens `myscalablepolicy` aus einem ElastiCache-for-Redis-Cluster namens `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"
}
```

Nutzung von AWS CloudFormation für Auto-Scaling-Richtlinien

Dieser Ausschnitt zeigt, wie eine Zielverfolgungsrichtlinie erstellt und auf eine [AWS::ElastiCache::ReplicationGroup](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalableTarget](#)-Ressource angewendet wird. 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, legen Sie geplante Aktionen an, die ElastiCache for Redis anweisen, Skalierungen zu bestimmten Zeitpunkten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie einen vorhandenen ElastiCache-for-Redis-Cluster an, wann die Skalierungsaktivität ausgeführt werden soll sowie welche minimale und maximale Kapazität zugelassen werden soll. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Sie können nur eine geplante Aktion für ElastiCache-for-Redis-Cluster erstellen, die bereits vorhanden ist. 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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis 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 dem **Aktionen**-Dropdown 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. Für **Wiederholung**, wählen Sie **Wiederkehrende** aus.
11. Für **Häufigkeit** wählen Sie den entsprechenden Wert aus.
12. Für **Startdatum und Beginnzeit** wählen 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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis 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 dem **Aktionen**-Dropdown 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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis 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 dem **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 folgenden APIs für die automatische Skalierung der Anwendung:

- [put-scheduled-action](#)

- [describe-scheduled-actions](#)
- [delete-scheduled-action](#)

Verwenden Sie AWS CloudFormation für die Erstellung einer geplanten Aktion

Dieser Ausschnitt zeigt, wie eine Zielverfolgungsrichtlinie erstellt und auf eine [AWS::ElastiCache::ReplicationGroup](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalableTarget](#)-Ressource angewendet wird. 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

Im Folgenden finden Sie Details zur Zielverfolgung und geplanten Richtlinien und zum Anwenden dieser Richtlinien mithilfe der 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 for Redis Auto Scaling erstellt und verwaltet die CloudWatch-Alarme, die die

Skalierungsrichtlinie auslösen, und berechnet die Skalierungsanpassung basierend auf der Metrik und dem Zielwert. 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 AVG EngineCPU-Auslastungsschwellenwert, der über alle Replikate aggregiert wird, die ElastiCache for Redis zum Auslösen eines auto-scaling gangs verwendet. Sie können das Auslastungsziel zwischen 35 Prozent und 70 Prozent festlegen.

Wenn der Dienst erkennt, dass Ihre `ElastiCacheReplicaEngineCPUUtilization`-Metrik gleich oder größer als die Zieleinstellung ist, werden Replikate in Ihren Shards automatisch erhöht. ElastiCache for Redis skaliert Ihre Clusterreplikate um eine Anzahl, die der größeren von zwei Zahlen entspricht: prozentuale Variation von Target und ein Replikat. Beim Scale-In kann ElastiCache for Redis nicht automatisch skaliert werden, es sei denn, der Gesamtwert liegt unter 75 Prozent des definierten Ziels.

Wenn Sie für ein Scale-Out-Beispiel jeweils 5 Shards und 1 Replikat haben:

Wenn Ihr Ziel um 30 Prozent verletzt, skaliert ElastiCache for Redis um 1 Replikat (max. (0,3, Standard 1)) über alle Shards hinweg. Dies führt zu 5 Shards mit jeweils 2 Replikaten

Wenn Sie für ein Scale-In-Beispiel Zielwert von 60 Prozent ausgewählt haben, wird ElastiCache for Redis nicht automatisch skalieren, bis die Metrik kleiner oder gleich 45 Prozent ist (25 Prozent unter dem Ziel 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 Ziel-Nachverfolgung verwenden, um eine horizontale Skalierung nach oben vorzunehmen, wenn die angegebene Metrik unter dem Zielwert liegt.

ElastiCache für Redis skaliert Replikate um maximal (% Abweichung von Target abgerundet, Standard 1) vorhandener Replikate über alle Shards im Cluster.

- 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. Der Grund hierfür ist, dass ElastiCache for Redis Auto Scaling stets vorsichtig agiert, indem beim Ermitteln der hinzuzufügenden oder zu entfernenden Kapazität Auf- oder Abrundungen vorgenommen werden. 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 Zielverfolgung-Skalierungsrichtlinien für einen ElastiCache-for-Redis-Cluster haben, vorausgesetzt, dass jede von ihnen eine andere Metrik verwendet. Die Absicht von ElastiCache for Redis Auto Scaling ist es, immer die Verfügbarkeit zu priorisieren, 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 for Redis Auto Scaling für eine Skalierungsrichtlinie für die Zielverfolgung verwaltet. ElastiCache for Redis Auto Scaling löscht die Alarme automatisch, wenn Sie die Skalierungsrichtlinie löschen oder den Cluster löschen.
- ElastiCache for Redis Auto Scaling hindert Sie nicht daran, Replicas manuell über Shards zu ändern. Diese manuellen Anpassungen haben keinerlei Auswirkungen auf bestehende CloudWatch Alarme, die an die Skalierungsrichtlinie angehängt sind, aber sie können sich auf Metriken auswirken, die diese CloudWatch-Alarme auslösen können.
- Diese CloudWatch Alarme, die von Auto Scaling verwaltet werden, werden über die AVG-Metrik für alle Shards im Cluster definiert. So kann Hot-Shards zu einem beliebigen Szenario führen:
 - Skalierung, wenn nicht erforderlich, da einige Hot Shards geladen werden, die einen CloudWatch Alarm auslösen
 - nicht skalieren, wenn dies erforderlich ist, aufgrund aggregierter AVG über alle Shards, die den Alarm nicht verletzen.

- ElastiCache for Redis-Standardlimits für Knoten pro Cluster gelten weiterhin. 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 Ihrer VPC genügend ENIs (Elastic Network Interfaces) zur Verfügung stehen, die während der Skalierung erforderlich sind. Weitere Informationen finden Sie unter [Elastic-Network-Schnittstellen](#).
- Wenn nicht genügend Kapazität von EC2 verfügbar ist, wird ElastiCache for Redis Auto Scaling erst skaliert, wenn die Kapazität verfügbar ist oder wenn Sie den Cluster manuell in die Instance-Typen ändern, die über genügend Kapazität verfügen.
- ElastiCache for Redis Auto Scaling unterstützt keine Skalierung von Replikaten mit einem Cluster mit `ReservedMemoryPercent` weniger als 25 Prozent. Weitere Informationen finden Sie unter [Reservierten Speicher verwalten](#).

Hinzufügen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie hinzufügen, indem Sie die verwenden AWS Management Console.

Hinzufügen einer Skalierungsrichtlinie mithilfe des AWS Management Console

Um eine Auto Scaling-Richtlinie zu einer ElastiCache für 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 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 (Richtliniename) einen Namen für diese Richtlinie ein.
8. FürSkalierbare Dimensionauf und danachReplicas-Dialogfeld aus erstellen.
9. Geben Sie für den Zielwert den durchschnittlichen Prozentsatz der CPU-Auslastung ein, den Sie auf ElastiCache Replicas beibehalten möchten. Dieser Wert muss größer 35 und kleiner 70 sein.

- Repliken 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 Verwaltung die ElastiCache for Redis Auto Scaling Scaling-Richtlinie erforderlich ist.
 12. Geben Sie für Maximale Kapazität die maximale Anzahl von Replikaten ein, für deren Verwaltung die ElastiCache for Redis 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 die AWS CLI oder die Application Auto Scaling API verwenden. Der erste Schritt besteht darin, Ihre ElastiCache for Redis-Replikationsgruppe bei ElastiCache for Redis Auto Scaling zu registrieren.

Bevor Sie Auto Scaling ElastiCache für Redis mit einem ElastiCache for Redis-Cluster verwenden können, registrieren Sie Ihren Cluster bei ElastiCache for Redis Auto Scaling. Sie tun dies, um die Skalierungsdimension und die Grenzwerte zu definieren, die auf diesen Cluster angewendet werden sollen. ElastiCache for Redis Auto Scaling skaliert den ElastiCache for Redis-Cluster dynamisch entlang der `elasticache:replication-group:Replicas` skalierbaren Dimension, die die Anzahl der Cluster-Replikat pro Shard darstellt.

Verwenden 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 der eindeutige Bezeichner beispielsweise der Name des ElastiCache For-Redis-Clusters. `replication-group/myscalablecluster`
- `--scalable-dimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.

- `--min-capacity` — Die Mindestanzahl von Replikaten, die ElastiCache für Redis 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](#).
- `--max-capacity` — Die maximale Anzahl von Replikaten, die ElastiCache für Redis Auto Scaling verwaltet werden sollen. 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 For-Redis-Cluster 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 ^  

```

Verwenden der 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 der eindeutige Bezeichner beispielsweise der Name des ElastiCache For-Redis-Clusters. `replication-group/myscalablecluster`

- **ScalableDimension** — Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.
- **MinCapacity** — Die Mindestanzahl von Replikaten, die ElastiCache für Redis 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 für Redis Auto Scaling verwaltet werden sollen. 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 For-Redis-Cluster, der `myscalecluster` mit der Application Auto Scaling Scaling-API benannt ist. 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:Replicas",
  "MinCapacity": 1,
  "MaxCapacity": 5
}
```

Definieren einer Skalierungsrichtlinie

Eine Konfiguration der Skalierungsrichtlinie für die Zielverfolgung wird durch einen JSON-Block repräsentiert, in dem die Metriken und Zielwerte definiert sind. Sie können die Konfiguration einer 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 unter [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für 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](#)
- [Nutzung von AWS CloudFormation für Auto-Scaling-Richtlinien](#)
- [Geplante Skalierung](#)

Verwenden einer vordefinierten Metrik

Eine Konfiguration der Skalierungsrichtlinie für die Zielverfolgung wird durch einen JSON-Block repräsentiert, in dem die Metriken und Zielwerte definiert sind. Sie können die Konfiguration einer 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 unter [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für 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](#)
- [Anwenden einer Skalierungsrichtlinie für einen ElastiCache for Redis-Cluster](#)

Verwenden einer vordefinierten Metrik

Durch die Verwendung vordefinierter Metriken können Sie schnell eine Zielverfolgung-Skalierungsrichtlinie für einen ElastiCache-for-Redis-Cluster definieren, die mit der Zielverfolgung in

ElastiCache for Redis Auto Scaling arbeitet. Aktuell unterstützt ElastiCache für Redis die folgende vordefinierte Metrik in ElastiCache Replicas Auto Scaling:

`ElastiCacheReplicaEngineCPUUtilization`— Der Mittelwert der Metrik `EngineCPUUtilization` Utility in CloudWatch über alle Replikate im ElastiCache for Redis-Cluster. Der Durchschnittswert der `EngineCPUUtilization`-Metrik in CloudWatch über alle Replikate im ElastiCache for Redis-Cluster hinweg. Sie finden den aggregierten Metrikwert in CloudWatch unter `ElastiCache for RedisReplicationGroupId`, `Role` für die erforderliche `ReplicationGroupId` und Rollenreplikate.

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 basierend auf einer beliebigen ElastiCache-for-Redis-Metrik definieren, die sich proportional zur Skalierung ändert. Nicht alle ElastiCache-für-Redis-Metriken funktionieren 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-for-Redis-Cluster basierend auf einer durchschnittlichen CPU-Auslastung von 50 Prozent über alle Replikate in einem Cluster namens `my-db-cluster` an.

```
{"TargetValue": 50,
  "CustomizedMetricSpecification":
  {"MetricName": "EngineCPUUtilization",
    "Namespace": "AWS/ElastiCache",
    "Dimensions": [
      {"Name": "ReplicationGroup", "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 unter [TargetTrackingScalingPolicyConfiguration](#) der API-Referenz für 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 ElastiCache-for-Redis-Cluster basierend auf einer durchschnittlichen CPU-Auslastung von 40 Prozent über alle Replikate in diesem Cluster hinweg 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 durch die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung verhindern, dass in Ihrem ElastiCache-for-Redis-Cluster ein Scale-In erfolgt, 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 unter [TargetTrackingScalingPolicyConfiguration](#) in der API-Referenz für Application Auto Scaling.

Example

Das folgende Beispiel beschreibt die Konfiguration einer Zielverfolgung für eine Skalierungsrichtlinie. In dieser Konfiguration passt die vordefinierte Metrik `ElastiCacheReplicaEngineCPUUtilization` einen ElastiCache-for-Redis-Cluster basierend auf einer durchschnittlichen CPU-Auslastung von 40 Prozent über alle Replikate in diesem Cluster hinweg an. Die Konfiguration deaktiviert die Scale-In-Aktivität für die Skalierungsrichtlinie.

```
{
  "TargetValue": 40.0,
  "PredefinedMetricSpecification":
    {
      "PredefinedMetricType": "ElastiCacheReplicaEngineCPUUtilization"
    },
  "DisableScaleIn": true
}
```

Anwenden einer Skalierungsrichtlinie für einen ElastiCache for Redis-Cluster

Nach der Registrierung Ihres Clusters mit ElastiCache for Redis Auto Scaling und der Definition einer Skalierungsrichtlinie wenden Sie die Skalierungsrichtlinie auf den registrierten Cluster an. Um eine Skalierungsrichtlinie auf einen ElastiCache-for-Redis-Cluster anzuwenden, können Sie die AWS CLI oder die Application-Auto-Scaling-API verwenden.

Verwendung der AWS CLI

Um eine Skalierungsrichtlinie auf Ihren ElastiCache-for-Redis-Cluster anzuwenden, verwenden Sie den [put-scaling-policy](#)-Befehl mit den folgenden Parametern:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--policy-type` – Setzen Sie diesen Wert auf `TargetTrackingScaling`.
- `—resource-id` — Die Ressourcenkennung für den ElastiCache for Redis-Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner ist der Name des ElastiCache-for-Redis-Clusters, zum Beispiel `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 ElastiCache for Redis-Cluster verwendet werden soll.

Example

Im folgenden Beispiel wenden Sie eine Zielverfolgungs-Skalierungsrichtlinie namens `myscalablepolicy` zu einem ElastiCache für Redis-Cluster namens `myscalablecluster` mit ElastiCache für Redis Auto Scaling. 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 der API

Um eine Skalierungsrichtlinie auf Ihren ElastiCache-for-Redis-Cluster über die Application-Auto-Scaling-API anzuwenden, verwenden Sie die [PutScalingPolicy](#)-API-Operation mit den folgenden Parametern:

- `PolicyName` – Der Name der Skalierungsrichtlinie.
- `PolicyType` – Setzen Sie diesen Wert auf `TargetTrackingScaling`.
- `ResourceId` — Die Ressourcenkennung für den ElastiCache for Redis-Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und der eindeutige Bezeichner ist der Name des ElastiCache-for-Redis-Clusters, zum Beispiel `replication-group/myscalablecluster`.
- `ServiceNamespace` — Legen Sie diesen Wert auf fest `elasticache`.
- `ScalableDimension` – Setzen Sie diesen Wert auf `elasticache:replication-group:Replicas`.
- `TargetTrackingScalingPolicyConfiguration` – Die Konfiguration der Skalierungsrichtlinie für die Zielverfolgung, die für den ElastiCache-for-Redis-Cluster verwendet werden soll.

Example

Im folgenden Beispiel wenden Sie eine Zielverfolgungs-Skalierungsrichtlinie namens `myscalablepolicy` zu einem ElastiCache for Redis-Cluster namens `myscalablecluster` mit ElastiCache für Redis Auto Scaling. 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 über die AWS Management Console, die AWS CLI oder die Application-Auto-Scaling-API bearbeiten.

Bearbeiten einer Skalierungsrichtlinie über die AWS Management Console

Sie können Richtlinien mit dem Typ Vordefinierte Metriken nur bearbeiten, indem Sie die AWS Management Console

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich 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 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 über die AWS CLI oder die Application-Auto-Scaling-API

Über die AWS CLI oder die Application-Auto-Scaling-API können Sie eine Skalierungsrichtlinie auf die gleiche Weise bearbeiten, wie Sie sie anwenden:

- Wenn Sie die Application Auto Scaling API verwenden, geben Sie den Namen der zu bearbeitenden Richtlinie im Parameter `PolicyName` an. Legen Sie neue Werte für die Parameter fest, die Sie ändern möchten.

Weitere Informationen finden Sie unter [Anwenden einer Skalierungsrichtlinie für einen ElastiCache for Redis-Cluster](#).

Löschen einer Skalierungsrichtlinie

Sie können eine Skalierungsrichtlinie mit der AWS Management Console, der AWS CLI oder der Application Auto Scaling API löschen

Löschen einer Skalierungsrichtlinie mithilfe der 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 Redis aus
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 der AWS CLI oder der Application Auto Scaling API

Sie können die AWS CLI oder die Application Auto Scaling API verwenden, um eine Skalierungsrichtlinie aus einem ElastiCache Cluster zu löschen.

CLI

Verwenden Sie den [delete-scaling-policy](#)Befehl mit den folgenden Parametern, um eine Skalierungsrichtlinie aus Ihrem ElastiCache for Redis-Cluster zu löschen:

- `--policy-name` – Der Name der Skalierungsrichtlinie.
- `--resource-id` — Die Ressourcen-ID für den for Redis-Cluster. ElastiCache Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des ElastiCache 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 Zielverfolgungs-Skalierungsrichtlinie namens `myscalablepolicy` aus einem ElastiCache-Cluster namens `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 API-Vorgang [DeleteScalingPolicy](#) Application Auto Scaling mit den folgenden Parametern, um eine Skalierungsrichtlinie aus Ihrem ElastiCache for Redis-Cluster zu löschen:

- `PolicyName` — Der Name der Skalierungsrichtlinie.
- `ResourceId` — Die ResourceID für den ElastiCache for Redis-Cluster. Für diesen Parameter ist der Ressourcentyp `ReplicationGroup` und die eindeutige Kennung beispielsweise der Name des ElastiCache Clusters. `replication-group/myscalablecluster`
- `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 ElastiCache for Redis-Cluster benannt ist, der `myscalablecluster` mit der Application Auto Scaling API benannt ist.

```

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"
}

```

Nutzung von AWS CloudFormation für Auto-Scaling-Richtlinien

Dieser Ausschnitt zeigt, wie eine geplante Aktion erstellt und auf eine [AWS::ElastiCache::ReplicationGroup](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalableTarget-Ressource](#) angewendet wird. 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, legen Sie geplante Aktionen an, die ElastiCache for Redis anweisen, Skalierungen zu bestimmten Zeitpunkten durchzuführen. Wenn Sie eine geplante Aktion erstellen, geben Sie einen vorhandenen ElastiCache-for-Redis-Cluster an, wann die Skalierungsaktivität ausgeführt werden soll sowie welche minimale und maximale Kapazität zugelassen werden soll. Sie können geplante Aktionen erstellen, die nur einmal skalieren oder wiederholt geplant ausgeführt werden.

Sie können nur eine geplante Aktion für ElastiCache-for-Redis-Cluster erstellen, die bereits vorhanden ist. 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 folgenden APIs für die automatische Skalierung der Anwendung:

- [put-scheduled-action](#)
- [describe-scheduled-actions](#)

- [delete-scheduled-action](#)

Verwenden von AWS CloudFormation So erstellen Sie Auto Scaling Richtlinien

Dieser Ausschnitt zeigt, wie eine geplante Aktion erstellt und auf eine [AWS::ElastiCache::ReplicationGroup](#)-Ressource unter Verwendung der [AWS::ApplicationAutoScaling::ScalableTarget-Ressource](#) angewendet wird. 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

Redis ist eine verteilte In-Memory-Datenbank, die Sharding und Replikation unterstützt. Cluster von ElastiCache für Redis sind die verteilte Implementierung von Redis, mit der Daten auf mehrere Redis-Knoten partitioniert werden können. Ein Cluster von ElastiCache für Redis hat zwei Betriebsmodi: Clustermodus aktiviert (CME) und Clustermodus deaktiviert (CMD). Im CME-Modus arbeitet Redis als verteilte Datenbank mit mehreren Shards und Knoten, während Redis im CMD-Modus als einzelner Knoten funktioniert.


Vor der Migration von CMD zu CME müssen die folgenden Bedingungen 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 Redis-Client, der das Cluster-Protokoll nutzen kann, und einen Konfigurationsendpunkt verwenden.
- Das automatische Failover muss auf dem Cluster mit mindestens einem Replikat aktiviert sein.
- Die für die Migration erforderliche Mindestversion der Redis-Engine ist 7.0.

Um von CMD zu CME zu migrieren, muss die Konfiguration des Clustermodus vom deaktivierten in den 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 für Redis automatisch die entsprechende Standardparametergruppe mit einer clusterfähigen Konfiguration aus. Der clusterfähige Parameterwert ist für einen CMD-Cluster auf no festgelegt. 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 Engine-Parametern unter Verwendung von Parametergruppen](#)

1. Vorbereiten – Erstellen Sie einen Test-CME-Cluster und stellen Sie sicher, dass Ihr Stack bereit ist, damit zu arbeiten. ElastiCache für Redis hat keine Möglichkeit, die Bereitschaft zu überprüfen. Weitere Informationen finden Sie unter [Erstellen eines Clusters](#).
2. Die bestehende CMD-Clusterkonfiguration so ändern, dass sie mit dem Clustermodus kompatibel ist – In diesem Modus wird ein einzelner Shard bereitgestellt und ElastiCache für Redis funktioniert nicht nur als einzelner Knoten, sondern 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 Redis-Clusterprotokoll und den Konfigurationsendpunkt verwenden können. Gehen Sie wie folgt vor, um den Redis-Clustermodus in einen kompatiblen Clustermodus zu ändern:

 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 ein Clustermodus-Parameter in der [ModifyReplicationGroup](#)-Anfrage definiert wird.

- a. Bei Verwendung der AWS Management Console finden Sie weitere Informationen unter [Ändern einer Replikationsgruppe](#). Legen Sie den Cluster-Modus auf Kompatibel fest.
- b. Bei Verwendung der API finden Sie weitere Informationen unter [ModifyReplicationGroup](#). Aktualisieren Sie den Parameter `ClusterMode` auf `compatible`.
- c. Bei Verwendung der AWS CLI finden Sie weitere Informationen unter [modify-replication-group](#). Aktualisieren Sie den Parameter `cluster-mode` auf `compatible`.

Nachdem der Redis-Clustermodus in den kompatiblen Clustermodus geändert wurde, gibt die [DescribeReplicationGroups](#)-API den Cluster-Konfigurationsendpunkt von ElastiCache für Redis zurück. 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 ermitteln](#).

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 Redis-Clients zur Verwendung des Clusterprotokolls migriert sind und dass der Konfigurationsendpunkt des Clusters nicht in Gebrauch ist.

- a. Bei Verwendung der AWS Management Console finden Sie weitere Informationen unter [Ändern einer Replikationsgruppe](#). Legen Sie den Cluster-Modus auf Aktiviert fest.
- b. Bei Verwendung der API finden Sie weitere Informationen unter [ModifyReplicationGroup](#). Aktualisieren Sie den Parameter `ClusterMode` auf `enabled`.
- c. Bei Verwendung der AWS CLI finden Sie weitere Informationen unter [modify-replication-group](#). Aktualisieren Sie den Parameter `cluster-mode` auf `enabled`.

Nachdem der Clustermodus auf aktiviert geändert wurde, werden die Endpunkte gemäß der Redis-Clusterspezifikation konfiguriert. Die [DescribeReplicationGroups](#)-API gibt den Clustermodus-Parameter `enabled` und die Cluster-Endpunkte zurück, die jetzt von Anwendungen verwendet werden können, um eine Verbindung mit dem Cluster herzustellen.

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, vom kompatiblen Clustermodus (CMD) zum deaktivierten Clustermodus (CMD) zurückzukehren und die ursprünglichen Konfigurationen beizubehalten.

Ändern der Clusterkonfiguration vom kompatiblen in den deaktivierten Clustermodus

1. Bei Verwendung der AWS Management Console finden Sie weitere Informationen unter [Ändern einer Replikationsgruppe](#). Legen Sie den Cluster-Modus auf Deaktiviert fest.
2. Bei Verwendung der API finden Sie weitere Informationen unter [ModifyReplicationGroup](#). Aktualisieren Sie den Parameter `ClusterMode` auf `disabled`.
3. Bei Verwendung der AWS CLI finden Sie weitere Informationen unter [modify-replication-group](#). Aktualisieren Sie den Parameter `cluster-mode` auf `disabled`.

Nachdem der Clustermodus auf deaktiviert geändert wurde, gibt die [DescribeReplicationGroups](#)-API den Clustermodus-Parameter `disabled` zurück.

AWS Regionsübergreifende Replikation mithilfe globaler Datenspeicher

Note

Der globale Datenspeicher ist derzeit nur für selbst entworfene Cluster verfügbar.

Mithilfe der Funktion Global Datastore for Redis können Sie mit einer vollständig verwalteten, schnellen, zuverlässigen und sicheren Replikation zwischen Regionen arbeiten. AWS Mit dieser Funktion können Sie regionsübergreifende Read Replica-Cluster ElastiCache für Redis erstellen, um Lesevorgänge mit geringer 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 \(Konsole\)](#)
- [Verwenden 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 befinden als der primäre Cluster.

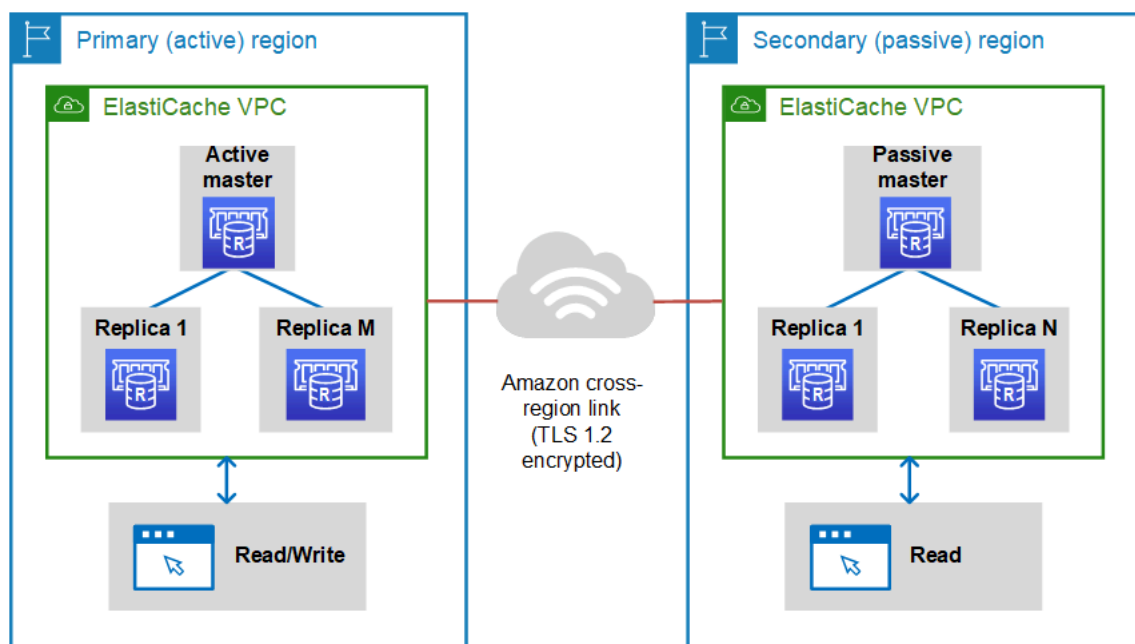
Wenn Sie einen globalen Datenspeicher in erstellen ElastiCache, repliziert ElastiCache for Redis Ihre Daten automatisch vom primären Cluster auf den sekundären Cluster. Sie wählen die AWS Region

aus, in der die 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 Redis bietet folgende 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, die älter als m5 oder r5 sind, nicht unterstützt.
- Sie können die Replikation für einen primären Cluster von einer AWS Region zu einem sekundären Cluster in bis zu zwei anderen Regionen einrichten. AWS

Note

Eine Ausnahme sind die Regionen China (Peking) und China (Ningxia), wo eine Replikation nur zwischen den beiden Regionen erfolgen kann.

- Mit globalen Datenspeichern können Sie nur in VPC-Clustern arbeiten. Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#). Globale Datenspeicher werden nicht unterstützt, wenn Sie EC2-Classic verwenden. Weitere Informationen finden Sie unter [EC2-Classic](#) im Amazon EC2 EC2-Benutzerhandbuch.

Note

Derzeit können Sie keine globalen Datenspeicher in [Verwenden von Local Zones mit ElastiCache](#) verwenden.

- ElastiCache unterstützt kein automatisches Failover von einer Region in eine andere. AWS 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 für Redis](#).
- Globale Datenspeicher unterstützen [Verschlüsselung im Ruhezustand](#), [Verschlüsselung während der Übertragung](#) und [Redis AUTH](#).
- Globale Datenspeicher unterstützen Internet Protocol Version 6 (IPv6) nicht.
- 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-Version muss Redis 5.0.6 oder höher sein.

2. Fügen Sie bis zu zwei sekundäre Cluster in verschiedenen AWS Regionen hinzu, wiederum mit der Redis 5.0.6-Engine oder höher.

In den folgenden Verfahren erfahren Sie, wie Sie einen globalen Datenspeicher für Redis erstellen und andere Operationen mit der for Redis-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 die Redis-Engine 5.0.6 oder höher verwenden.

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 Konsole unter https://console.aws.amazon.com/elasticache/ ElastiCache](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 Customer Managed AWS KMS key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwenden von vom Kunden verwalteten AWS KMS-Schlüsseln](#).

- Verschlüsselung während der Übertragung – Aktiviert die Verschlüsselung von Daten bei der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Redis-Engine Version 6.0 und höher die Verschlüsselung während

der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Optionen für die Zugriffskontrolle 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 [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
- Redis AUTH Standardbenutzer - Ein Authentifizierungsmechanismus für den Redis-Server. Weitere Informationen finden Sie unter [Redis 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 Replikat
- Subnetzgruppe
- Bevorzugte Availability Zone(s)
- Sicherheitsgruppen
- Vom Kunden verwaltet (AWS KMS-Schlüssel)
- Redis-AUTH-Token
- Aktivieren automatischer Sicherungen.
- Aufbewahrungszeitraum für Backups
- Backup-Fenster
- Wartungsfenster
- Thema für SNS-Benachrichtigung

8. Wählen Sie Erstellen. 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 Redis-Seite wird aktualisiert, um anzugeben, ob ein Cluster Teil eines globalen Datenspeichers ist, einschließlich:

- 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 [Redis-spezifische Parameter](#) und [Erstellen einer Parametergruppe](#).

 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 (▼)). 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](#).

- e. Wenn Sie einen Redis-Cluster (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 Redis-Cluster (Cluster-Modus aktiviert) erstellen:
 - i. Wählen Sie unter Anzahl der Shards die Anzahl der Shards (Partitionen/ Knotengruppen), die Sie für diesen Redis-Cluster (Cluster-Modus aktiviert) wünschen.

Für einige Versionen von Redis (Cluster-Modus aktiviert), können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern:

- Redis 3.2.10 und höher – Wenn Ihr Cluster mit Redis 3.2.10 oder höheren Versionen arbeitet, können Sie die Anzahl der Shards in Ihrem Cluster dynamisch ändern. Weitere Informationen finden Sie unter [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#).
 - Andere Redis-Versionen – Wenn Ihr Cluster eine Version von Redis vor Version 3.2.10 verwendet, gibt es einen anderen Ansatz. 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.

Für Redis (Cluster-Modus aktiviert) gelten die folgenden Einschränkungen.

- 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 [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 standardmäßige IPv4-Subnetzgruppe bereit, oder Sie können wählen, ob Sie eine neue erstellen möchten. Für IPv6 müssen Sie eine Subnetzgruppe mit einem IPv6-CIDR-Block erstellen. Wenn Sie sich für dual stack (Dual-Stack) entscheiden, müssen Sie einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

Weitere Informationen finden Sie unter [Erstellen eines Subnetzes in Ihrer 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 Availability Zones](#).

The screenshot shows the configuration interface for an Amazon ElastiCache Redis cluster. At the top, there are two dropdown menus: 'Slots and keyspaces' set to 'Custom distribution' and 'Availability zone(s)' set to 'Specify availability zones'. Below these is a table with three columns: 'Slots/Keyspaces', 'Primary', and 'Replica 1'. The table has three rows for 'NodeGroup 1', 'NodeGroup 2', and 'NodeGroup 3'. In the 'Slots/Keyspaces' column, the value '0-1234' is circled in red. In the 'Primary' column, the value 'us-east-1b' is circled in red. The 'Replica 1' column shows 'us-east-1a' for all three node groups.

| | Slots/Keyspaces | Primary | Replica 1 |
|-------------|-----------------|------------|------------|
| NodeGroup 1 | 0-1234 | us-east-1a | us-east-1a |
| NodeGroup 2 | | us-east-1b | us-east-1a |
| NodeGroup 3 | | us-east-1a | us-east-1a |


Angeben von Keyspaces und Availability Zones

8. Wählen Sie Next (Weiter).
9. Unter Advanced Redis settings (Erweiterte Redis-Einstellungen):
 - 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 Customer Managed AWS KMS key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwendung von kundenverwalteten Schlüsseln aus 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 bei Redis-Engine Version 6.0 und höher die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Optionen für die Zugriffskontrolle 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 [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
 - Redis AUTH Standardbenutzer - Ein Authentifizierungsmechanismus für den Redis-Server. Weitere Informationen finden Sie unter [Redis AUTH](#).
 - Redis AUTH – Ein Authentifizierungsmechanismus für Redis-Server. Weitere Informationen finden Sie unter [Redis AUTH](#).

 Note

Für Redis-Versionen ab 3.2.6, ausgenommen Version 3.2.10, ist Redis AUTH 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 Ihre VPC](#) im Benutzerhandbuch zu Amazon VPC.

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 zu Redis-Sicherungen und -Wiederherstellungen finden Sie unter [Snapshot und Wiederherstellung](#).


11. (Optional) Geben Sie ein Wartungsfenster an. Das Wartungsfenster ist die Zeit, in der Regel eine Stunde lang, 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 Uhrzeiten sind in UCT angegeben.

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

12. (Optional) Für Protokolle:
 - Unter `Protokollformat` wählen Sie entweder `Text` oder `JSON` aus.
 - 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 [Markieren Ihrer ElastiCache-Ressourcen](#).
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 Customer Managed AWS KMS key und dann den Schlüssel auswählen. Weitere Informationen finden Sie unter [Verwenden von vom Kunden verwalteten AWS KMS-Schlüsseln](#).

- Verschlüsselung während der Übertragung – Aktiviert die Verschlüsselung von Daten bei der Übertragung. Weitere Informationen finden Sie unter [Verschlüsselung während der Übertragung](#). Wenn Sie für Redis Engine Version 6.4 die Verschlüsselung während der Übertragung aktivieren, werden Sie aufgefordert, eine der folgenden Optionen für die Zugriffskontrolle 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 [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#).
- Redis AUTH Standardbenutzer - Ein Authentifizierungsmechanismus für den Redis-Server. Weitere Informationen finden Sie unter [Redis AUTH](#).

 Note

Für Redis-Versionen zwischen 4.0.2, als die Verschlüsselung bei der Übertragung erstmals unterstützt wurde, und 6.0.4, ist Redis AUTH 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 (AWS KMS-Schlüssel)
 - Redis-AUTH-Token
 - Aktivieren automatischer Sicherungen.
 - Aufbewahrungszeitraum für Backups
 - Backup-Fenster
 - Wartungsfenster
 - Thema für SNS-Benachrichtigung
19. Wählen Sie Erstellen. 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 Redis-Seite wird außerdem aktualisiert, um anzugeben, ob ein Cluster Teil eines globalen Datenspeichers ist, einschließlich der folgenden Elemente:

- 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 https://console.aws.amazon.com/elasticache/ 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.
- Redis Engine Version: Die Redis Engine-Version, die den globalen Datenspeicher ausführt

- 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
- Redis AUTH: 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 Redis mit aktiviertem Clustermodus zeigt die Verzögerung die maximale Verzögerung 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/.](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 Customer Managed AWS KMS key und dann den Schlüssel auswählen.

- Verschlüsselung während der Übertragung
 - Redis 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)
 - Redis-AUTH-Token
 - Aktivieren automatischer Sicherungen.
 - Aufbewahrungszeitraum für Backups
 - Backup-Fenster
 - Wartungsfenster
 - Thema für 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 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 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
 - Modify engine version (Engine-Version ändern) – Es ist nur Redis-Engine-Version 5.0.6 oder höher 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 zu einem der regionalen ElastiCache Replikate. Weitere Informationen finden Sie im Abschnitt [Automatisches Failover](#).

Für Redis-Cluster mit aktiviertem Cluster-Modus:

- 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. Verwenden Sie die Redis-Konsole oder die API-Operation, um die Parametergruppe dieses Clusters zu ändern.

[ModifyCacheCluster](#) Weitere Informationen finden Sie unter [Modifizieren einer 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 die [ResetCacheParameterGroup](#) API-Operation, um eine gesamte Parametergruppe oder bestimmte Parameter zurückzusetzen.

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 ElastiCache heraufstufen, garantiert Redis nicht, 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/.](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/.](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 aus.

Der globale Datenspeicher wechselt zum Status Deleting (Löschen).

Verwenden globaler Datenspeicher (CLI)

Das AWS Command Line Interface (AWS CLI) ermöglicht die Steuerung mehrerer AWS-Services über die Befehlszeile und automatisiert diese mithilfe von Skripten. Sie können die AWS-CLI für (einmalige) Ad-hoc-Operationen verwenden.

Herunterladen und Konfigurieren des AWS CLI

Die AWS CLI kann auf Windows, macOS oder Linux ausgeführt werden. Gehen Sie folgendermaßen vor, um sie herunterzuladen und zu konfigurieren.

So laden Sie den CLI herunter, installieren und konfigurieren ihn

1. Laden Sie die AWS-CLI auf der [AWS Command Line Interface](#)-Webseite herunter.
2. Befolgen Sie die Anweisungen zum Installieren der AWS-CLI und Konfigurieren der AWS-CLI im AWS Command Line Interface Benutzerhandbuch.

Verwenden der 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 ElastiCache wendet beim Erstellen automatisch ein Präfix auf die globale Datenspeicher-ID an. 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 werden alle AWS-Regionen und ihr globales Datenspeicher-ID-Präfix gelistet.

| Regionsname/Region | Präfix |
|---|--------|
| Region USA Ost (Ohio) us-east-2 | fpkhr |
| Region USA Ost (Nord-Virginia) us-east-1 | ldgnf |
| Region US West (N. California) us-west-1 | virxk |
| Region USA West (Oregon) us-west-2 | sgaui |

| Regionsname/Region | Präfix |
|---|--------|
| Region Kanada (Zentral) ca-central-1 | bxodz |
| Region Asien-Pazifik (Mumbai) ap-south-1 | erpgt |
| Region Asien-Pazifik (Tokio) ap-northeast-1 | qusw |
| Asia Pacific (Seoul) Region ap-northeast-2 | lfqnh |
| Region Asien-Pazifik (Osaka) ap-northeast-3 | nlapn |
| Region Asien-Pazifik (Singapur) ap-southeast-1 | v1qxn |
| Region Asien-Pazifik (Sydney) ap-southeast-2 | vbgxd |
| Region Europa (Frankfurt) eu-central-1 | iudkw |
| Region Europa (Irland) eu-west-1 | gxeiz |
| Region Europa (London) eu-west-2 | okuqm |

| Regionsname/Region | Präfix |
|--|--------|
| Region Europa (Paris) eu-west-3 | fgjhi |
| Region Südamerika (São Paulo) sa-east-1 | juxlw |
| Region China (Peking) cn-north-1 | emvgo |
| Region China (Ningxia) cn-northwest-1 | ckbem |
| Region Asien-Pazifik (Hongkong) ap-east-1 | knjmp |
| AWS GovCloud (USA-West) us-gov-west-1 | sgwui |

- [create-replication-group](#) – Verwenden Sie diese Operation, um sekundäre Cluster für einen globalen Datenspeicher zu erstellen, indem Sie den Namen des globalen Datenspeichers für den `--global-replication-group-id`-Parameter angeben.

```
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 diese Operation aufgerufen und ein `--global-replication-group-id`-Wert übergeben wird, leitet ElastiCache for Redis die Werte aus der primären Replikationsgruppe der globalen Replikationsgruppe für die folgenden Parameter ab. 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
```

- [delete-global-replication-group](#)

```
aws elasticache delete-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --retain-primary-replication-group defaults to true
```

- [disassociate-global-replication-group](#)

```
aws elasticache disassociate-global-replication-group \  
  --global-replication-group-id my global datastore \  
  --replication-group-id my secondary cluster \  
  --replication-group-region the AWS Region in which the secondary cluster resides
```

- [failover-global-replication-group](#)

```
aws elasticache failover-replication-group \  
  --global-replication-group-id my global datastore \  
  --primary-region The AWS Region of the primary cluster \  
  --primary-replication-group-id The name of the global datastore, including the  
  suffix.
```

- [increase-node-groups-in-global-replication-group](#)

```
aws elasticache increase-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [decrease-node-groups-in-global-replication-group](#)

```
aws elasticache decrease-node-groups-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name \  
  --node-group-count 3
```

- [rebalance-shards-in-global-replication-group](#)

```
aws elasticache rebalance-shards-in-global-replication-group \  
  --apply-immediately yes \  
  --global-replication-group-id global-replication-group-name
```

Verwenden Sie die Hilfe, um alle verfügbaren Befehle von ElastiCache for Redis aufzulisten.

```
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

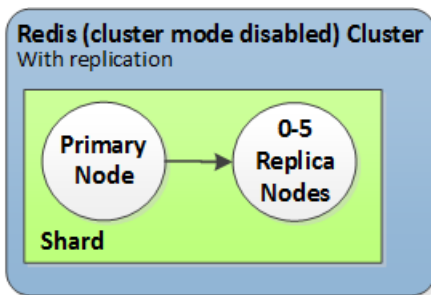
Amazon ElastiCache Redis-Cluster mit einem Knoten sind In-Memory-Einheiten mit eingeschränkten Datenschutzdiensten (AOF). Sollte Ihr Cluster aus irgendeinem Grund ausfallen, verlieren Sie alle Daten des Clusters. Wenn Sie jedoch die Redis-Engine ausführen, können Sie 2 bis 6 Knoten zu einem Cluster mit Replikaten gruppieren, wobei 1 bis 5 schreibgeschützte Knoten Replikatdaten 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 Grafik zu sehen ist, ist die Replikationsstruktur in einem Shard (in der API/CLI als Knotengruppe bezeichnet) enthalten, der in einem Redis-Cluster enthalten ist. Redis-Cluster (Cluster-Modus deaktiviert) haben immer einen Shard. Redis-Cluster (Cluster-Modus aktiviert) können bis zu 500 Shards haben, wobei die Daten des Clusters über die Shards partitioniert 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 auf maximal 500 pro Cluster erhöht werden, wenn die Version der Redis-Engine 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. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht 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.



Redis-Cluster (Cluster-Modus deaktiviert) verfügen über 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 [Minimieren von Ausfällen mit Redis](#).

Themen

- [Grundlegendes zur Redis-Replikation](#)
- [Replikation: Redis \(Cluster-Modus deaktiviert\) im Vergleich zu Redis \(Cluster-Modus aktiviert\)](#)
- [Minimierung der Ausfallzeiten ElastiCache bei Redis mit Multi-AZ](#)
- [So werden Synchronisation und Backup implementiert](#)
- [Erstellen einer Redis-Replikationsgruppe](#)
- [Anzeigen der Details einer Replikationsgruppe](#)
- [Suchen von Endpunkten von Replikationsgruppen](#)
- [Ändern einer Replikationsgruppe](#)
- [Löschen einer Replikationsgruppe](#)
- [Ändern der Anzahl von Replikaten](#)
- [Redis \(Cluster-Modus deaktiviert\) -Replikationsgruppen](#)

Grundlegendes zur Redis-Replikation

Redis implementiert die Replikation auf zwei Arten:

- Mit einem einzigen Shard, der alle Daten des Clusters in jedem Knoten enthält – Redis (Clustermodus deaktiviert)
- Mit Daten, die auf bis zu 500 Shards partitioniert sind – Redis (Clustermodus aktiviert)

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 Version der Redis-Engine 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. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht 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.

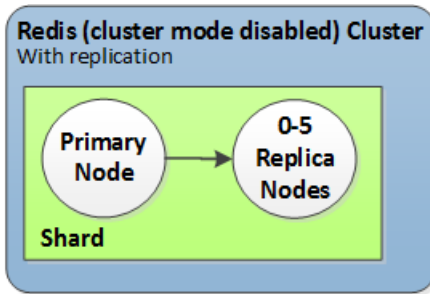
Themen

- [Redis \(Clustermodus deaktiviert\)](#)
- [Redis \(Cluster-Modus aktiviert\)](#)

Redis (Clustermodus deaktiviert)

Ein Redis-Cluster (Clustermodus deaktiviert) verfügt über einen einzelnen Shard, in dem sich eine Sammlung von Redis-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.



Redis-Cluster (Cluster-Modus deaktiviert) mit einem einzelnen Shard und Replikatknoten

Sie können Redis-Cluster (Clustermodus deaktiviert) mit Replikatknoten verwenden, um Ihre Redis-Lösung für leseintensive Anwendungen ElastiCache zu skalieren oder um eine große Anzahl von Clients zu unterstützen, die gleichzeitig aus demselben Cluster lesen.

Alle Knoten in einem Redis-Cluster (Cluster-Modus 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 die Schreibausfallzeit zu reduzieren, aktivieren Sie Multi-AZ mit automatischem Failover für Ihren Redis-Cluster (Clustermodus deaktiviert) mit Replikaten. Weitere Informationen finden Sie unter [Minimierung der Ausfallzeiten ElastiCache bei Redis mit Multi-AZ](#).

Sie können die Rollen der Knoten innerhalb des Redis-Clusters (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 [Redis \(Cluster-Modus deaktiviert\) -Replikationsgruppen](#).

Redis (Cluster-Modus aktiviert)

Ein Redis-Cluster (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.

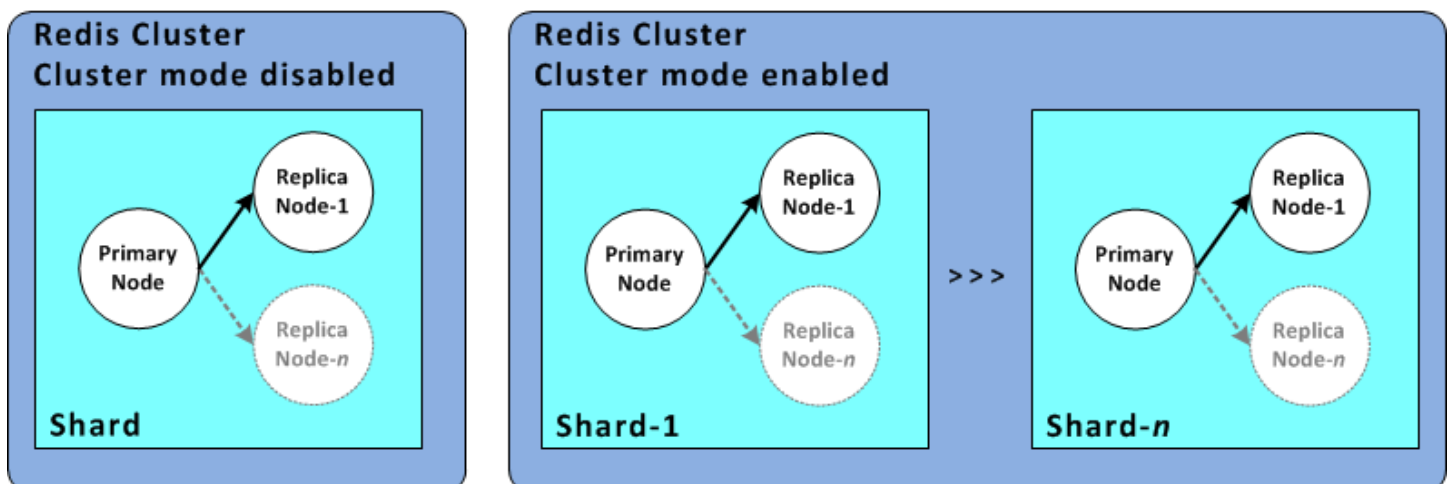
Das Knoten- oder Shard-Limit kann auf maximal 500 pro Cluster erhöht werden, wenn die Version der Redis-Engine 5.0.6 oder höher ist. Sie können beispielsweise einen Cluster mit 500 Knoten konfigurieren, der zwischen 83 Shards (ein primärer Knoten und 5 Replikat pro Shard) und 500 Shards (ein primärer Knoten und keine Replikat) umfasst. Stellen Sie sicher, dass für die Erhöhung genügend IP-Adressen verfügbar sind. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht 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 über die Shards in einem Redis-Cluster (Clustermodus aktiviert) partitioniert.

Anwendungen verwenden den Konfigurations-Endpunkt des Redis-Clusters (Clustermodus aktiviert), um eine Verbindung mit den Knoten im Cluster herzustellen. Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#).



Redis-Cluster (Cluster-Modus aktiviert) mit mehreren Shards und Replikatknoten

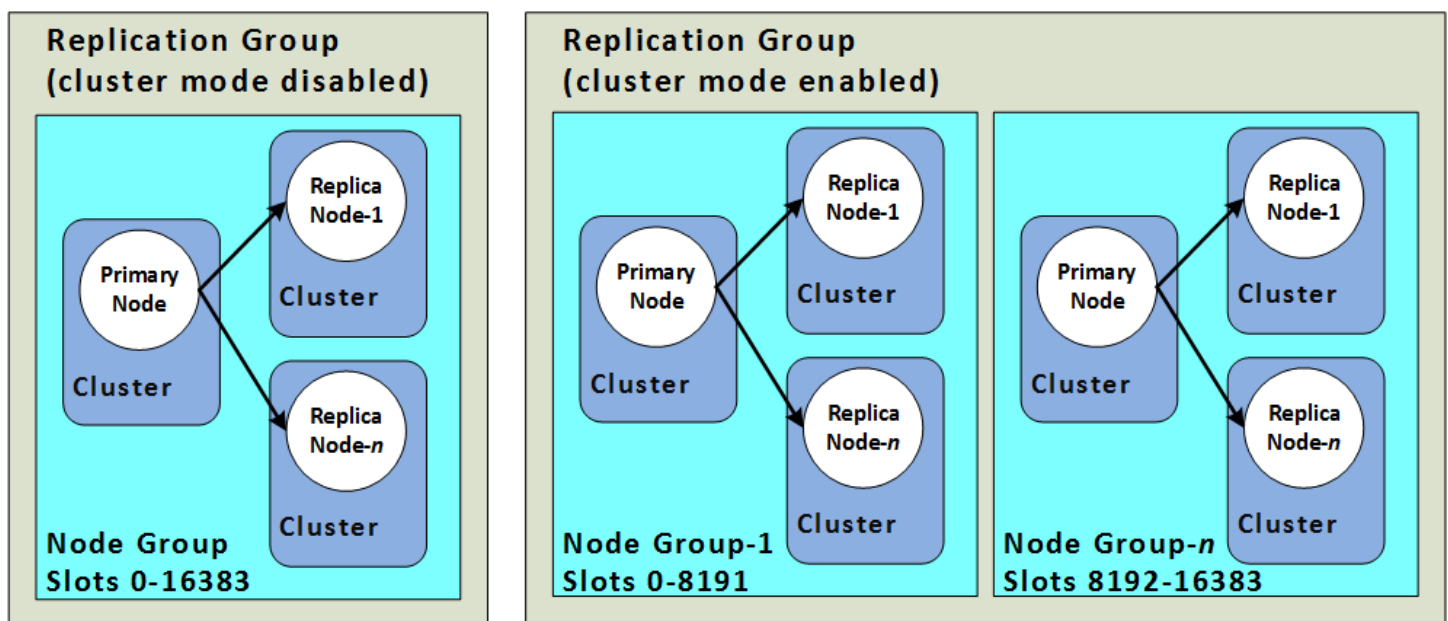
Alle Knoten in einem Redis-Cluster (Cluster-Modus aktiviert) müssen sich in derselben Region befinden. Zur Verbesserung der Fehlertoleranz können Sie sowohl primären Knoten als auch Read Replicas in mehreren Availability Zones innerhalb dieser Region bereitstellen.

Derzeit gibt es in Redis (Cluster-Modus aktiviert) einige Beschränkungen.

- Keine der Replikatknoten können zum primären Knoten manuell heraufgestuft werden.

Replikation: Redis (Cluster-Modus deaktiviert) im Vergleich zu Redis (Cluster-Modus aktiviert)



Ab Redis-Version 3.2 können Sie eine von zwei unterschiedlichen Redis-Clustertypen (API/CLI: Replikationsgruppen) erstellen. Ein Redis-Cluster (Clustermodus deaktiviert) verfügt immer über einen einzelnen Shard (API/CLI: Knotengruppe) mit bis zu 5 Read Replica-Knoten. Ein Redis-Cluster (Clustermodus aktiviert) verfügt über bis zu 500 Shards mit jeweils 1 bis 5 Read Replica-Knoten.



Redis-Cluster (Cluster-Modus deaktiviert) und Redis-Cluster (Cluster-Modus aktiviert)

Die folgende Tabelle fasst wichtige Unterschiede zwischen Redis-Clustern (Clustermodus deaktiviert) und Redis-Clustern (Clustermodus aktiviert) zusammen.

Vergleich von Redis-Clustern (Cluster-Modus deaktiviert) und Redis-Clustern (Cluster-Modus aktiviert)

| Funktion | Redis-Cluster (Clustermodus deaktiviert) | Redis (Cluster-Modus aktiviert) |
|----------------------|--|---|
| Anpassbar | Ja. Unterstützt das Hinzufügen und Löschen von Replikatknoten und das Hochskalieren des Knotentyps. | Begrenzt. Weitere Informationen finden Sie unter Engine-Versionen und Upgrades und Skalieren von Clustern in Redis (Clustermodus aktiviert) . |
| Datenpartitionierung | Nein | Ja |
| Shards | 1 | 1 bis 500 |
| Read Replicas | 0 bis 5  Important Wenn Sie bei einem Knotenausfall über keine Replikate verfügen, gehen alle Ihre Daten verloren. | 0 bis 5 pro Shard.  Important Wenn Sie bei einem Knotenausfall über keine Replikate verfügen, gehen alle Ihre Daten in diesem Shard verloren. |
| Multi-AZ | Ja, mit mindestens 1 Replikat. Optional. Standardmäßig aktiviert | Ja Optional. Standardmäßig aktiviert |
| Snapshots (Backups) | Ja, es wird eine einzelne .rdb-Datei erstellt. | Ja, es wird für jeden Shard eine eindeutige .rdb-Datei erstellt. |
| Wiederherstellung | Ja, mit einer einzelnen .rdb-Datei aus einem Redis-Cluster (Clustermodus deaktiviert). | Ja, mit .rdb-Dateien von einem Redis-Cluster (Clustermodus deaktiviert) oder einem Redis- |

| | | |
|-----------------------|---|---|
| Funktion | Redis-Cluster (Clustermodus deaktiviert) | Redis (Cluster-Modus aktiviert) |
| | | Cluster (Clustermodus aktiviert). |
| Unterstützt von | Allen Redis-Versionen | Redis 3.2 und höher |
| Engine aktualisierbar | Ja, mit einigen Einschränkungen. Weitere Informationen finden Sie unter Engine-Versionen und Upgrades . | Ja, mit einigen Einschränkungen. Weitere Informationen finden Sie unter Engine-Versionen und Upgrades . |
| Verschlüsselung | Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis-Versionen) sowie 4.0.10 und höher. | Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis-Versionen) sowie 4.0.10 und höher. |
| HIPAA-konform | Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis-Versionen) sowie 4.0.10 und höher. | Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis-Versionen) sowie 4.0.10 und höher. |
| Zu PCI DSS konform | Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis-Versionen) sowie 4.0.10 und höher. | Versionen 3.2.6 (für EOL geplant, siehe Zeitplan für das Ende der Lebensdauer der Redis-Versionen) sowie 4.0.10 und höher. |
| Online-Resharding | N/A | Version 3.2.10 (geplant für EOL, siehe Zeitplan für das Ende der Lebensdauer der Redis-Versionen) und höher. |

Welchen Cluster soll ich wählen?

Berücksichtigen Sie bei der Wahl zwischen Redis (Clustermodus deaktiviert) oder Redis (Clustermodus aktiviert) die folgenden Faktoren:

- Skalierung im Vergleich zu Partitionierung – Geschäftliche Anforderungen ändern sich. Sie müssen entweder für Spitzenbedarf vorsorgen oder bei Bedarfsänderungen skalieren. Redis (Cluster-Modus deaktiviert) unterstützt 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 [Skalieren von Redis-Clustern \(Clustermodus deaktiviert\) mit Replikatknoten](#).

Redis (Clustermodus aktiviert) unterstützt die Partitionierung Ihrer Daten über 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 [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#).

- Knotengröße vs. Anzahl der Knoten – Da ein Redis-Cluster (Clustermodus deaktiviert) nur über einen Shard verfügt, muss der Knotentyp groß genug sein, um alle Daten des Clusters plus den erforderlichen Overhead aufzunehmen. Andererseits können die Knotentypen kleiner sein, obwohl Sie mehr davon benötigen, da Sie Ihre Daten bei Verwendung eines Redis-Clusters (Clustermodus aktiviert) auf mehrere Shards partitionieren können. Weitere Informationen finden Sie unter [Auswahl der Knotengröße](#).
- Reads vs. Writes – Wenn die primäre Last auf Ihrem Cluster das Lesen von Daten durch Anwendungen ist, können Sie einen Redis-Cluster (Clustermodus deaktiviert) skalieren, indem Sie Read Replicas hinzufügen und löschen. Es gibt jedoch maximal 5 Read Replicas. Wenn die Last auf Ihrem Cluster schreibintensiv ist, können Sie von den zusätzlichen Schreibendpunkten eines Redis-Clusters (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 der Ausfallzeiten ElastiCache bei Redis mit Multi-AZ

Es gibt eine Reihe von Fällen, in denen ElastiCache Redis möglicherweise einen Primärknoten austauschen 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 verbreitet auch den DNS-Namen (Domain Name Service) des beworbenen Replikats. 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:

- ElastiCache Denn bei Redis Cluster werden die geplanten Knotenersetzungen abgeschlossen, während der Cluster eingehende Schreibanforderungen bearbeitet.
- Für Cluster, bei denen der Redis-Clustermodus deaktiviert und Multi-AZ aktiviert ist und die auf Version 5.0.6 oder einer späteren Version der Engine ausgeführt werden, erfolgt der geplante Knotenaustausch, während der Cluster eingehende Schreibanforderungen ausführt.
- Für Cluster, bei denen der Redis-Clustermodus deaktiviert und Multi-AZ aktiviert ist und die auf Version 4.0.10 oder einer früheren Version der Engine ausgeführt werden, ist bei DNS-Aktualisierungen möglicherweise eine kurze Schreibunterbrechung zu beobachten. 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 AWS CLI, der oder der API aktivieren. ElastiCache

Die Aktivierung von ElastiCache Multi-AZ auf Ihrem Redis-Cluster (in der API und CLI, Replikationsgruppe) 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 Redis-Clustern unterstützt, die mehr als einen Knoten in jedem Shard haben.

Themen

- [Aktivieren von Multi-AZ](#)
- [Fehlerszenarien mit Multi-AZ-Antworten](#)
- [Testen des automatischen Failovers](#)
- [Einschränkungen bei Redis Multi-AZ](#)

Aktivieren von Multi-AZ

Sie können Multi-AZ aktivieren, wenn Sie einen Cluster (API oder CLI, Replikationsgruppe) mithilfe der ElastiCache Konsole oder der ElastiCache API erstellen oder ändern. AWS CLI

Sie können Multi-AZ nur auf Redis-Clustern (Clustermodus deaktiviert) aktivieren, die über mindestens eine verfügbare Read Replica (Lesereplikat) verfügen. Cluster ohne Read Replicas bieten keine hohe Verfügbarkeit oder Fehlertoleranz. Weitere Informationen zum Erstellen eines Clusters mit Replikation finden Sie unter [Erstellen einer Redis-Replikationsgruppe](#). Weitere Informationen zum Hinzufügen einer Read Replica zu einem Cluster mit Replikation finden Sie unter [Hinzufügen eines Lesereplikats für Redis-Replikationsgruppen \(Clustermodus deaktiviert\)](#).

Themen

- [Aktivieren von Multi-AZ \(Konsole\)](#)
- [Aktivieren von Multi-AZ \(AWS CLI\)](#)
- [Multi-AZ \(API\) ElastiCache aktivieren](#)

Aktivieren von Multi-AZ (Konsole)

Sie können Multi-AZ mithilfe der ElastiCache Konsole aktivieren, wenn Sie einen neuen Redis-Cluster erstellen oder indem Sie einen vorhandenen Redis-Cluster mit Replikation ändern.

Multi-AZ ist standardmäßig in Redis-Clustern (Cluster-Modus 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 [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#). 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“ [Mit dem AWS Management Console](#).

Aktivieren von Multi-AZ (AWS CLI)

Das folgende Codebeispiel verwendet die 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 in etwa folgendermaßen aus.

```
{  
  "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.

Multi-AZ (API) ElastiCache aktivieren

Im folgenden Codebeispiel wird die ElastiCache API verwendet, um Multi-AZ für die Replikationsgruppe zu aktivieren. `redis12`

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 API-Referenz zu diesen Themen:

- [CreateCacheCluster](#)
- [CreateReplicationGruppe](#)
- [ModifyReplicationGruppe](#)

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 primären 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. ElastiCacheverbreitet den DNS-Namen des beworbenen Replikats.

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:

- [Suche nach den Endpunkten eines Redis-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Suchen der Endpunkte für Replikationsgruppen \(AWS CLI\)](#)
- [Finden von Endpunkten für 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 verbreitet den DNS-Namen des beworbenen 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:

- [Suche nach den Endpunkten eines Redis-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Suchen der Endpunkte für Replikationsgruppen \(AWS CLI\)](#)
- [Finden von Endpunkten für 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:

- [Suche nach den Endpunkten eines Redis-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Suchen der Endpunkte für Replikationsgruppen \(AWS CLI\)](#)
- [Finden von Endpunkten für 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 den automatischen Failover aktiviert haben, können Sie ihn mithilfe der ElastiCache Konsole, der AWS CLI, der 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 (in der ElastiCache API als Knotengruppen bezeichnet, in der AWS CLI) in einem beliebigen Zeitraum von 24 Stunden zu testen.
- Wenn Sie diese Operation für Shards in verschiedenen Clustern (in der API und CLI als Replikationsgruppen bezeichnet) aufrufen, können diese Aufrufe gleichzeitig erfolgen.
- In einigen Fällen können Sie diesen Vorgang mehrmals auf verschiedenen Shards in derselben Redis-Replikationsgruppe (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:

- [Anzeigen von ElastiCache-Ereignissen](#) im ElastiCache -Benutzerhandbuch
- [DescribeEvents](#) in der ElastiCache -API-Referenz
- [describe-events](#) in der AWS CLI -Befehlsreferenz.
- Diese API wurde entwickelt, um das Verhalten Ihrer Anwendung im Falle eines ElastiCache Failovers zu testen. Sie wurde nicht als Betriebstool zum Einleiten eines Failovers konzipiert, um ein Problem mit dem Cluster zu beheben. Darüber hinaus kann diese API unter bestimmten Bedingungen, z. B. bei großen Betriebsereignissen, blockiert werden.

Themen

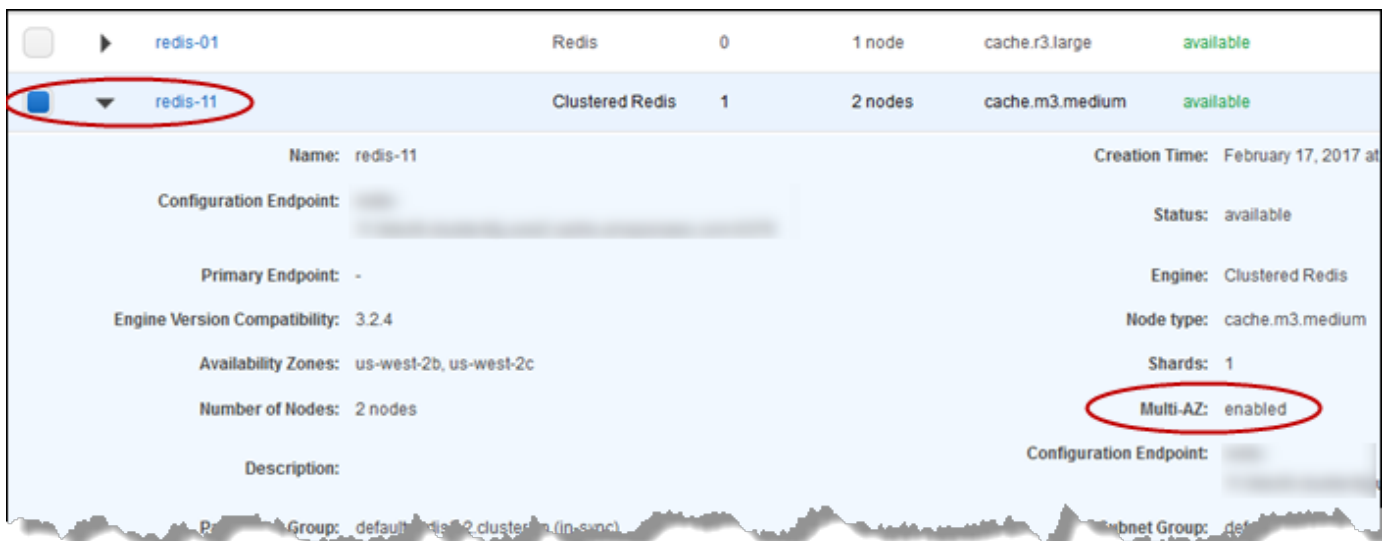
- [Testen des automatischen Failovers mit dem AWS Management Console](#)
- [Testen des automatischen Failovers mit dem AWS CLI](#)
- [Testen Sie den automatischen Failover mithilfe der API ElastiCache](#)

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 Redis aus.
3. Aktivieren Sie in der Liste der Redis-Cluster das Kontrollkästchen links neben einem Cluster, 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 [Mit dem AWS Management Console](#).



5. Wählen Sie für Redis (Cluster-Modus deaktiviert) den Namen des Clusters aus.

Gehen Sie für Redis (Clustermodus aktiviert) wie folgt vor:

- a. Wählen Sie den Cluster-Namen aus.

- b. Wählen Sie auf der Seite Shards für den Shard (in API und CLI als Knotengruppe bezeichnet), auf dem das Failover getestet werden soll, 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 des automatischen Failovers 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 Redis-Cluster (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 mithilfe der API ElastiCache

Mithilfe der ElastiCache API-Operation können Sie das automatische Failover auf jedem Cluster testen, für den 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 ElastiCache API-Referenz
- [DescribeEvents](#) in der ElastiCache API-Referenz

Einschränkungen bei Redis Multi-AZ

Beachten Sie die folgenden Einschränkungen für Redis Multi-AZ:

- Multi-AZ wird auf Redis Version 2.8.6 und höher unterstützt.
- Redis Multi-AZ wird auf T1-Knotentypen nicht unterstützt.
- Die Redis-Replikation läuft asynchron ab. 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 primären Replikat heraufgestuft werden soll, wird ElastiCache für Redis das Replikat mit 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 Lesereplikate auf Redis manuell auf primär hochstufen (Clustermodus deaktiviert), ist dies nur möglich, 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 Redis-Konsole tun, indem Sie das Kontrollkästchen Auto Failover für die Replikationsgruppe deaktivieren. Sie können dies mithilfe von `tun`, AWS CLI indem Sie die `AutomaticFailoverEnabled` Eigenschaft `false` beim Aufrufen der Operation auf setzen. `ModifyReplicationGroup`
 3. Stufen Sie das Lesereplikat zum primären Knoten herauf.
 4. Multi-AZ wieder aktivieren.
- ElastiCache für Redis schließen sich Multi-AZ und Append-Only File (AOF) 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 Redis und gilt nicht nur für Multi-AZ. ElastiCache Weitere Informationen zu diesem Redis-Verhalten finden Sie unter [Replikation](#) auf der Redis-Website.

Wichtig

Für Redis Version 2.8.22 und höher können keine externen Replikate erstellt werden. Für Redis-Versionen vor 2.8.22 empfehlen wir, kein externes Redis-Replikat mit einem Multi-AZ-fähigen For-Redis-Cluster zu verbinden. ElastiCache Diese nicht unterstützte Konfiguration kann zu Problemen führen, die eine ordnungsgemäße Durchführung von Failover und Wiederherstellung verhindern. ElastiCache Um ein externes Redis-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 Redis ermöglichen Sicherung und Synchronisierung zwischen dem primären und den Replikatknoten. Die Methode, mit der Sicherung und Synchronisierung implementiert werden, ist jedoch je nach der Redis-Version verschieden.

Redis Version 2.8.22 und höher

Redis-Replikationen ab Version 2.8.22 wählen zwischen zwei Sicherungsmethoden. Weitere Informationen finden Sie unter [Redis-Versionen 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 vor 2.8.22

Redis-Sicherungen und -Synchronisierungen vor Version 2.8.22 wurden in drei Schritten durchgeführt.

1. Vergabeln und serialisieren Sie die Clusterdaten als Hintergrundprozess auf dem Datenträger. Dadurch wird ein zeitpunktbezogener Snapshot erstellt.
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 Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#).

3. Übertragen Sie schließlich die Cachedaten und anschließend das Änderungsprotokoll auf den Replikatknoten.

Erstellen einer Redis-Replikationsgruppe

Es bieten sich Ihnen die folgenden Optionen für das Erstellen eines Clusters mit Replikatknoten. Eine Option gilt, wenn Sie bereits über einen verfügbaren Redis-Cluster (Clustermodus deaktiviert) verfügen, der keinem Cluster zugeordnet ist, welcher über Replikate verfügt, die als Primärknoten verwendet werden sollen. Die andere trifft zu, wenn Sie einen Primärknoten mit dem Cluster und Lesereplikaten erstellen müssen. Derzeit muss ein Redis-Cluster (Clustermodus aktiviert) von Grund auf neu erstellt werden.

Option 1: [Erstellen einer Replikationsgruppe mit einem verfügbaren Redis-Cluster \(Clustermodus deaktiviert\)](#)

Verwenden Sie diese Option, um einen vorhandenen Redis-Cluster mit einem einzelnen Knoten (Clustermodus deaktiviert) 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 mit einem verfügbaren Redis-Cluster \(Clustermodus deaktiviert\)](#).

Important

Sie können keinen Redis-Cluster (Clustermodus aktiviert) mit einem vorhandenen Cluster erstellen. Informationen zum Erstellen eines Redis-Clusters (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole finden Sie unter [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Option 2: [Erstellen einer Redis-Replikationsgruppe von Grund auf](#)

Verwenden Sie diese Option, wenn Sie nicht bereits über einen verfügbaren Redis-Cluster (Clustermodus deaktiviert) verfügen, um ihn als Primärknoten des Clusters zu verwenden, oder wenn Sie einen Redis-Cluster (Clustermodus aktiviert) erstellen möchten. Siehe [Erstellen einer Redis-Replikationsgruppe von Grund auf](#).

Erstellen einer Replikationsgruppe mit einem verfügbaren Redis-Cluster (Clustermodus deaktiviert)

Ein verfügbarer Cluster ist ein vorhandener Redis-Einzelknoten-Cluster. Derzeit unterstützt Redis (Clustermodus aktiviert) nicht das Erstellen eines Clusters mit Replikaten unter Verwendung eines verfügbaren Einzelknoten-Clusters. Informationen zum Erstellen eines Redis-Clusters (Clustermodus aktiviert) finden Sie unter [Erstellen eines Redis-Clusters \(Clustermodus aktiviert\) \(Konsole\)](#).

Das folgende Verfahren kann nur angewandt werden, wenn Sie über einen Redis-Cluster (Clustermodus deaktiviert) mit einem einzelnen Knoten verfügen. Der Knoten dieses Clusters wird zum primären Knoten im neuen Cluster. Wenn Sie keinen Redis-Cluster (Clustermodus deaktiviert) haben, den Sie als primären Cluster des neuen Clusters verwenden können, lesen Sie [Erstellen einer Redis-Replikationsgruppe von Grund auf](#).

Erstellen einer Replikationsgruppe mit einem verfügbaren Redis-Cluster (Konsole)

Weitere Informationen finden Sie im Thema [Verwendung von AWS Management Console](#).

Erstellen einer Replikationsgruppe mit einem verfügbaren Redis-Cache-Cluster (AWS CLI)

Eine Replikationsgruppe mit Read Replicas wird anhand eines verfügbarer Redis-Cache-Clusters für den primären Cluster in zwei Schritten erstellt, wenn die AWS CLI verwendet wird.

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 `aws-elasticache 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. z. B. `sample-repl-group-001`.

Namensbeschränkungen für Replikationsgruppen in Redis (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 verfügbaren Knotens des Redis-Clusters (Clustermodus deaktiviert), der der Primärknoten in dieser Replikationsgruppe sein soll.

Der folgende Befehl erstellt die Replikationsgruppe `sample-repl-group` unter Verwendung des verfügbaren Redis-Clusters `redis01` (Clustermodus deaktiviert) als Primärknoten der Replikationsgruppe. 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 [AWS CLI Thema `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
```


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 Redis-Cluster (Cluster-Modus deaktiviert) (ElastiCache API)

Wenn Sie die ElastiCache API verwenden, 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. z. B. `sample-repl-group-001`.

Namensbeschränkungen für Replikationsgruppen in Redis (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.

ReplicationGroupBeschreibung

Beschreibung des Clusters mit Replikaten.

NumCacheCluster

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 Redis-Clusters (Clustermodus deaktiviert), der der Primärknoten in diesem Cluster sein soll.

Der folgende Befehl erstellt den Cluster mit Replikaten `sample-repl-group` unter Verwendung des verfügbaren Redis-Clusters (Clustermodus deaktiviert) `redis01` als Primärknoten der Replikationsgruppe. 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:

- [CreateReplicationGruppe](#)
- [ModifyReplicationGruppe](#)

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 `myRep1Group` 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 ElastiCache API-Thema [CreateCacheCluster](#).

Erstellen einer Redis-Replikationsgruppe von Grund auf

Im Folgenden finden Sie Informationen über das Erstellen einer Redis-Replikationsgruppe ohne Verwendung eines vorhandenen Redis-Clusters als primären Cluster. Sie können eine Redis-Replikationsgruppe (Clustermodus deaktiviert) oder Redis-Replikationsgruppe (Clustermodus aktiviert) von Grund auf neu erstellen, indem Sie die ElastiCache-Konsole, die AWS CLI oder die ElastiCache-API verwenden.

Bevor Sie fortfahren, entscheiden Sie, ob Sie eine Redis-Replikationsgruppe (Clustermodus deaktiviert) oder eine Redis-Replikationsgruppe (Clustermodus aktiviert) erstellen möchten. Hilfreiche Informationen zur Entscheidung finden Sie unter [Replikation: Redis \(Cluster-Modus deaktiviert\) im Vergleich zu Redis \(Cluster-Modus aktiviert\)](#).

Themen

- [Erstellen einer Redis-Replikationsgruppe \(Clustermodus deaktiviert\) von Grund auf](#)
- [Erstellen einer Replikationsgruppe in Redis \(Clustermodus aktiviert\) von Grund auf](#)

Erstellen einer Redis-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf

Sie können mithilfe der ElastiCache Konsole, der oder der ElastiCache API eine Redis-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen. AWS CLI Eine Redis-Replikationsgruppe (Clustermodus deaktiviert) hat immer eine Knotengruppe, einen primären Cluster und bis zu fünf Lesereplikate. Redis-Replikationsgruppen (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.

Um eine Redis-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu zu erstellen, gehen Sie wie folgt vor:

Erstellen einer Redis-Replikationsgruppe (Clustermodus deaktiviert) (AWS CLI)

Das folgende Verfahren erstellt eine Redis-Replikationsgruppe (Clustermodus deaktiviert) mithilfe der AWS CLI.

Wenn Sie eine Redis-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und alle ihre Knoten mit einem einzigen Aufruf des AWS CLI `create-replication-group` Befehls. Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe.

Namensbeschränkungen für Replikationsgruppen in Redis (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 den Leistungsdetails der einzelnen Knotentypen finden Sie unter [Amazon-EC2-Instance-Typen](#).

--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](#).

--cache-parameter-group

Geben Sie eine Parametergruppe an, die Ihrer Engine-Version entspricht. Wenn Sie Redis 3.2.4 oder höher ausführen, geben Sie die `default.redis3.2`-Parametergruppe oder eine von `default.redis3.2` abgeleitete Parametergruppe an, um eine Redis-Replikationsgruppe (Clustermodus deaktiviert) zu erstellen. Weitere Informationen finden Sie unter [Redis-spezifische Parameter](#).

--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.

- Ihre Replikationsgruppe muss Redis-Version 3.2.6 oder 4.0.10 ausführen.
- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter `--cache-subnet-group` einbinden.
- Sie müssen auch den Parameter `--auth-token` einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

Der folgende Vorgang erstellt eine Redis-Replikationsgruppe (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 AWS CLI Thema [create-replication-group](#).

Eine Redis-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen (API) ElastiCache

Mit dem folgenden Verfahren wird mithilfe der API eine Redis-Replikationsgruppe (Clustermodus deaktiviert) erstellt. ElastiCache

Wenn Sie eine Redis-Replikationsgruppe (Clustermodus deaktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und all ihre Knoten mit einem einzigen Aufruf der ElastiCache CreateReplicationGroup API-Operation. Schließen Sie die folgenden Parameter ein.

ReplicationGroupID

Der Name der zu erstellenden Replikationsgruppe.

Namensbeschränkungen für Replikationsgruppen in Redis (Clustermodus aktiviert) 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.

ReplicationGroupBeschreibung

Ihre Beschreibung der Replikationsgruppe.

NumCacheCluster

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 den Leistungsdetails der einzelnen Knotentypen finden Sie unter [Amazon-EC2-Instance-Typen](#).

--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](#).

CacheParameterGruppe

Geben Sie eine Parametergruppe an, die Ihrer Engine-Version entspricht. Wenn Sie Redis 3.2.4 oder höher ausführen, geben Sie die `default.redis3.2`-Parametergruppe oder eine von `default.redis3.2` abgeleitete Parametergruppe an, um eine Redis-Replikationsgruppe (Clustermodus deaktiviert) zu erstellen. Weitere Informationen finden Sie unter [Redis-spezifische Parameter](#).

--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.

- Ihre Replikationsgruppe muss Redis-Version 3.2.6 oder 4.0.10 ausführen.
- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter `CacheSubnetGroup` einbinden.
- Sie müssen auch den Parameter `AuthToken` einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

Der folgende Vorgang erstellt die Redis-Replikationsgruppe `myRep1Group` (Cluster-Modus deaktiviert) 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=myRep1Group  
&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](#).

Erstellen einer Replikationsgruppe in Redis (Clustermodus aktiviert) von Grund auf

Sie können einen Redis-Cluster (Clustermodus aktiviert) (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole, der oder der API AWS CLI erstellen. ElastiCache Eine Redis-Replikationsgruppe (Clustermodus aktiviert) verfügt über 1 bis 500 Shards (API/CLI: Knotengruppen), einen Primärknoten in jedem Shard und bis zu 5 Lesereplikate 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 Version der Redis-Engine 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. Häufige Fallstricke sind Subnetze in der Subnetzgruppe, die einen zu kleinen CIDR-Bereich haben, oder Subnetze, die gemeinsam genutzt und von anderen Clustern stark beansprucht 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 Redis (Clustermodus aktiviert)

- [Erstellen eines Redis-Clusters \(Clustermodus aktiviert\) \(Konsole\)](#)
- [Erstellen einer Redis-Replikationsgruppe \(Clustermodus aktiviert\) von Grund auf \(AWS CLI\)](#)
- [Eine Replikationsgruppe in Redis \(Clustermodus aktiviert\) von Grund auf neu erstellen \(ElastiCache API\)](#)

Erstellen eines Redis-Clusters (Clustermodus aktiviert) (Konsole)

Informationen zum Erstellen eines Redis-Clusters (Clustermodus aktiviert) finden Sie unter [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#). 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.

Erstellen einer Redis-Replikationsgruppe (Clustermodus aktiviert) von Grund auf (AWS CLI)

Das folgende Verfahren erstellt eine Redis-Replikationsgruppe (Clustermodus aktiviert) mithilfe der AWS CLI.

Wenn Sie eine Redis-Replikationsgruppe (Clustermodus aktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und all ihre Knoten mit einem einzigen AWS CLI `create-replication-group` Befehlsaufruf. Schließen Sie die folgenden Parameter ein.

`--replication-group-id`

Der Name der zu erstellenden Replikationsgruppe.

Namensbeschränkungen für Replikationsgruppen in Redis (Clustermodus aktiviert) 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.

`--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 den Leistungsdetails der einzelnen Knotentypen finden Sie unter [Amazon-EC2-Instance-Typen](#).

`--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](#).

--cache-parameter-group

Geben Sie die `default.redis6.x.cluster.on`-Parametergruppe oder eine von `default.redis6.x.cluster.on` abgeleitete Parametergruppe an, um eine Redis-Replikationsgruppe (Clustermodus aktiviert) zu erstellen. Weitere Informationen finden Sie unter [Parameteränderungen von Redis 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.

- Ihre Replikationsgruppe muss Redis-Version 3.2.6 oder 4.0.10 ausführen.

- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter `--cache-subnet-group` einbinden.
- Sie müssen auch den Parameter `--auth-token` einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

Die folgende Operation erstellt die Redis-Replikationsgruppe `sample-repl-group` (Cluster-Modus aktiviert) mit drei Knotengruppen/Shards (`--num-node-groups`), jeweils mit drei Knoten, einem primären und zwei Lesereplikaten (`--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 Redis-Replikationsgruppe (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, das 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 ElastiCache die Availability Zone 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 Redis-Replikationsgruppe (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 AWS CLI Thema. [create-replication-group](#)

Eine Replikationsgruppe in Redis (Clustermodus aktiviert) von Grund auf neu erstellen (ElastiCache API)

Das folgende Verfahren erstellt mithilfe der API eine Redis-Replikationsgruppe (Clustermodus aktiviert). ElastiCache

Wenn Sie eine Redis-Replikationsgruppe (Clustermodus aktiviert) von Grund auf neu erstellen, erstellen Sie die Replikationsgruppe und all ihre Knoten mit einem einzigen Aufruf der ElastiCache CreateReplicationGroup API-Operation. Schließen Sie die folgenden Parameter ein.

ReplicationGroupID

Der Name der zu erstellenden Replikationsgruppe.

Namensbeschränkungen für Replikationsgruppen in Redis (Clustermodus aktiviert) 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.

ReplicationGroupBeschreibung

Beschreibung der Replikationsgruppe.

NumNodeGruppen

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.

NodeGroupKonfiguration

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 ElastiCache die 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 den Leistungsdetails der einzelnen Knotentypen finden Sie unter [Amazon-EC2-Instance-Typen](#).

--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](#).

CacheParameterGruppe

Geben Sie die `default.redis6.x.cluster.on`-Parametergruppe oder eine von `default.redis6.x.cluster.on` abgeleitete Parametergruppe an, um eine Redis-Replikationsgruppe (Clustermodus aktiviert) zu erstellen. Weitere Informationen finden Sie unter [Parameteränderungen von Redis 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.

- Ihre Replikationsgruppe muss Redis-Version 3.2.6 oder 4.0.10 ausführen.
- Die Replikationsgruppe muss in einer Amazon VPC erstellt werden.
- Sie müssen außerdem den Parameter CacheSubnetGroup einbinden.
- Sie müssen auch den Parameter AuthToken einbinden, mit dem vom Kunden angegebenen Zeichenfolgenwert für Ihr AUTH-Token (Passwort), das für Operationen auf dieser Replikationsgruppe benötigt wird.

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, die AWS CLI für ElastiCache oder die ElastiCache-API verwenden. Der Konsolenprozess unterscheidet sich für Redis (Clustermodus deaktiviert) und Redis (Clustermodus aktiviert).

Anzeige der Details einer Replikationsgruppe

- [Anzeigen von Details für Redis \(Clustermodus deaktiviert\) mit Replikaten](#)
 - [Anzeigen von Details für eine Redis-Replikationsgruppe \(Clustermodus deaktiviert\) \(Konsole\)](#)
 - [Anzeigen von Details für eine Redis-Replikationsgruppe \(Clustermodus deaktiviert\) \(AWS CLI\)](#)

- [Details für eine Redis-Replikationsgruppe \(API\) \(Cluster-Modus deaktiviert\) anzeigen ElastiCache](#)
- [Anzeigen der Details einer Replikationsgruppe: Redis \(Clustermodus aktiviert\)](#)
 - [Anzeigen von Details für einen Redis-Cluster \(Cluster-Modus aktiviert\) \(Konsole\)](#)
 - [Anzeigen von Details für einen Redis-Cluster \(Clustermodus aktiviert\) \(AWS CLI\)](#)
 - [Details für einen Redis-Cluster \(Cluster Mode Enabled\) \(ElastiCache API\) anzeigen](#)
- [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#)
- [Anzeige der Details einer Replikationsgruppe \(ElastiCache-API\)](#)

Anzeigen von Details für Redis (Clustermodus deaktiviert) mit Replikaten

Sie können die Details eines Redis-Clusters (Clustermodus deaktiviert) mit Replikaten (API/CLI: Replikationsgruppe) mithilfe der ElastiCache Konsole, des AWS CLI for ElastiCache oder der API anzeigen. ElastiCache

Anzeigen der Details eines Redis-Clusters (Clustermodus deaktiviert)

- [Anzeigen von Details für eine Redis-Replikationsgruppe \(Clustermodus deaktiviert\) \(Konsole\)](#)
- [Anzeigen von Details für eine Redis-Replikationsgruppe \(Clustermodus deaktiviert\) \(AWS CLI\)](#)
- [Details für eine Redis-Replikationsgruppe \(API\) \(Cluster-Modus deaktiviert\) anzeigen ElastiCache](#)

Anzeigen von Details für eine Redis-Replikationsgruppe (Clustermodus deaktiviert) (Konsole)

Informationen zum Anzeigen der Details eines Redis-Clusters (Clustermodus deaktiviert) mit Replikaten mithilfe der ElastiCache Konsole finden Sie im Thema. [Anzeigen von Details eines Redis-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)

Anzeigen von Details für eine Redis-Replikationsgruppe (Clustermodus deaktiviert) (AWS CLI)

Ein AWS CLI Beispiel, in dem die Details einer Redis-Replikationsgruppe (Clustermodus deaktiviert) angezeigt werden, finden Sie unter [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#).

Details für eine Redis-Replikationsgruppe (API) (Cluster-Modus deaktiviert) anzeigen ElastiCache

Ein ElastiCache API-Beispiel, in dem die Details einer Redis-Replikationsgruppe (Clustermodus deaktiviert) angezeigt werden, finden Sie unter [Anzeige der Details einer Replikationsgruppe \(ElastiCache-API\)](#).

Anzeigen der Details einer Replikationsgruppe: Redis (Clustermodus aktiviert)

Anzeigen von Details für einen Redis-Cluster (Cluster-Modus aktiviert) (Konsole)

Informationen zum Anzeigen der Details eines Redis-Clusters (Clustermodus aktiviert) mithilfe der ElastiCache Konsole finden Sie unter [Anzeigen von Details für einen Redis-Cluster \(Cluster-Modus aktiviert\) \(Konsole\)](#).

Anzeigen von Details für einen Redis-Cluster (Clustermodus aktiviert) (AWS CLI)

Ein ElastiCache CLI-Beispiel, das die Details einer Redis-Replikationsgruppe (Clustermodus aktiviert) anzeigt, finden Sie unter [Anzeigen der Details einer Replikationsgruppe \(AWS CLI\)](#).

Details für einen Redis-Cluster (Cluster Mode Enabled) (ElastiCache API) anzeigen

Ein ElastiCache API-Beispiel, in dem die Details einer Redis-Replikationsgruppe (Clustermodus aktiviert) angezeigt werden, finden Sie unter [Anzeige der Details einer Replikationsgruppe \(ElastiCache-API\)](#).

Anzeigen der Details einer Replikationsgruppe (AWS CLI)

Sie können die Details einer Replikationsgruppe mit dem AWS CLI-Befehl `describe-replication-groups` 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 dieser Operation sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "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 unter AWS CLI im ElastiCache-Thema [describe-replication-groups](#).

Anzeige der Details einer Replikationsgruppe (ElastiCache-API)

Sie können die Details einer Replikation mit der AWS CLI-Operation `DescribeReplicationGroups` abrufen. 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 Referenzthema [DescribeReplicationGroups](#) zur ElastiCache-API .

Suchen von Endpunkten von Replikationsgruppen

Eine Anwendung kann eine Verbindung zu jedem beliebigen Knoten in einer Replikationsgruppe herstellen, vorausgesetzt, dass es über den DNS-Endpunkt und die Portnummer für den betreffenden Knoten verfügt. Je nachdem, ob Sie eine Redis-Replikationsgruppe (Clustermodus deaktiviert) oder eine Redis-Replikationsgruppe (Clustermodus aktiviert) ausführen, sind Sie an verschiedenen Endpunkten interessiert.

Redis (Clustermodus deaktiviert)

Redis-Cluster (Clustermodus deaktiviert) mit Replikaten haben drei Arten von Endpunkten: den primären Endpunkt, den Leser-Endpunkt und die Knotenendpunkte. Der primäre Endpunkt ist ein DNS-Name, der immer auf den 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 Reader-Endpunkt teilt eingehende Verbindungen zum Endpunkt gleichmäßig auf alle Read Replicas in einem ElastiCache for Redis-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-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-Eintrag, der nach dem Rotationsprinzip zu einer IP-Adresse eines der Replikationsknoten 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.

Redis (Clustermodus aktiviert)

Redis-Cluster (Clustermodus aktiviert) mit Replikaten haben eine andere Endpunktstruktur als Redis-Cluster (Clustermodus deaktiviert), da sie mehrere Shards (API/CLI: Knotengruppen) haben,

was bedeutet, dass sie auch mehrere Primärknoten haben. Redis (Cluster-Modus aktiviert) verfügt über einen Konfigurations-Endpunkt, der alle primären und Knotenendpunkte im Cluster „kennt“. Ihre Anwendung stellt eine Verbindung zum Konfigurationsendpunkt her. Jedes Mal, wenn Ihre Anwendung in den Konfigurationsendpunkt eines Clusters schreibt oder daraus liest, bestimmt Redis im Hintergrund, zu welchem Shard der Schlüssel gehört und welcher Endpunkt in betreffenden Shard zu verwenden ist. Dies läuft alles transparent für Ihre Anwendung ab.

Sie können die Endpunkte für einen Cluster mithilfe der ElastiCache Konsole, der oder der AWS CLI API finden. ElastiCache

Suchen von Replikationsgruppen-Endpunkten

Anweisungen zur Suche nach den Endpunkten für Ihre Replikationsgruppe finden Sie in einem der folgenden Themen:

- [Suche nach den Endpunkten eines Redis-Clusters \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Suchen von Endpunkten für einen Redis-Cluster \(Cluster-Modus aktiviert\) \(Konsole\)](#)
- [Suchen der Endpunkte für Replikationsgruppen \(AWS CLI\)](#)
- [Finden von Endpunkten für Replikationsgruppen \(ElastiCache-API\)](#)

Ändern einer Replikationsgruppe

Wichtige Einschränkungen

- ElastiCache unterstützt derzeit begrenzte Änderungen einer Redis-Replikationsgruppe (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 Redis-Cluster (Cluster-Modus aktiviert) mit dem API-Vorgang [ModifyReplicationGroupShardConfiguration](#) ändern (CLI: `modify-replication-group-shard-configuration`). Weitere Informationen finden Sie unter [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#).

Andere Änderungen an einem Redis-Cluster (Cluster-Modus aktiviert) erfordern, dass Sie einen Cluster mit dem neuen Cluster erstellen, der die Änderungen enthält.

- Sie können (Cluster-Modus deaktiviert) und Redis (Cluster-Modus aktiviert) -Cluster und -Replikationsgruppen 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 [Engine-Versionen und Upgrades](#).
- Sie können einen vorhandenen Cluster ElastiCache für Redis, der den Clustermodus deaktiviert verwendet, mithilfe der Konsole, der Gruppen-API oder des CLI-Befehls `ModifyReplicationmodify-replication-group` aktualisieren, um den Clustermodus aktiviert zu verwenden, wie im folgenden Beispiel gezeigt. Sie können auch den Schritten unter [Ändern des Clustermodus](#) folgen.

Sie können die Einstellungen eines Redis-Clusters (Clustermodus deaktiviert) mithilfe der ElastiCache Konsole, der oder der API ändern. AWS CLI ElastiCache ElastiCache unterstützt derzeit eine begrenzte Anzahl von Änderungen an einer Redis-Replikationsgruppe (Clustermodus aktiviert). Andere Änderungen erfordern die Erstellung eines Backups der aktuellen Replikationsgruppe und die Verwendung dieses Backups zum Seeden einer neuen Redis-Replikationsgruppe (Cluster-Modus aktiviert).

Themen

- [Unter Verwendung der AWS Management Console](#)

- [Mit dem AWS CLI](#)
- [Verwenden der API ElastiCache](#)

Unter Verwendung der AWS Management Console

Informationen zum Ändern eines Redis-Clusters (Cluster-Modus deaktiviert) finden Sie unter [Einen ElastiCache Cluster ändern](#).

Mit dem 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 von Multi-AZ für eine vorhandene 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 Redis-Clients sowohl über den aktivierten als auch den deaktivierten Clustermodus eine Verbindung herstellen. Nachdem Sie alle Redis-Clients zum aktivierten Clustermodus migriert haben, können Sie die Konfiguration des Clustermodus abschließen und den Clustermodus auf aktiviert festlegen.

Für Linux, macOS oder Unix:

Legen Sie den Clustermodus auf kompatibel fest.

```
aws elasticache modify-replication-group \  
  --multi-az-enabled
```



```
--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 [Modifying cluster mode](#) im ElastiCache for Redis User Guide.

Verwenden der API ElastiCache

Der folgende ElastiCache API-Vorgang aktiviert Multi-AZ für eine bestehende 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 zum ElastiCache ModifyReplicationGroup API-Vorgang finden Sie unter [ModifyReplicationGroup](#)

Löschen einer Replikationsgruppe

Wenn Sie einen Ihrer Cluster mit Replikaten (in API/CLI als Replikationsgruppen bezeichnet) nicht mehr benötigen, können Sie ihn löschen. Wenn Sie eine Replikationsgruppe löschen, löscht ElastiCache alle Knoten in der betreffenden Gruppe.

Sobald die Operation gestartet wurde, kann sie nicht mehr unterbrochen oder abgebrochen werden.

Warning

Wenn Sie einen ElastiCache-für-Redis-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.

Löschen einer Replikationsgruppe (Konsole)

Zum Löschen eines Clusters mit Replikaten informieren Sie sich unter [Löschen eines Clusters](#).

Löschen einer Replikationsgruppe (AWS CLI)

Mit dem Befehl [delete-replication-group](#) können Sie eine Replikationsgruppe 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 Parameter `ReplicationGroup` auf.

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 Redis-Replikationsgruppe mithilfe der AWS Management Console, der oder der AWS CLI API dynamisch erhöhen oder verringern. ElastiCache Wenn es sich bei Ihrer Replikationsgruppe um eine Redis-Replikationsgruppe (Cluster-Modus aktiviert) handelt, können Sie auswählen, welche Shards (Knotengruppen) die Anzahl der Replikate erhöhen oder verringern sollen.

Um die Anzahl von Replikaten in Ihrer Redis-Replikationsgruppe dynamisch zu ändern, wählen Sie aus der folgenden Tabelle den für Ihre Situation passenden Vorgang aus.

| So gehen Sie vor | Für Redis (Cluster-Modus aktiviert) | Für Redis (Cluster-Modus deaktiviert) |
|----------------------|--|--|
| Replikate hinzufügen | Erhöhen der Anzahl der Replikate in einem Shard | Erhöhen der Anzahl der Replikate in einem Shard Hinzufügen eines Lesereplikats für Redis-Replikationsgruppen (Clustermodus deaktiviert) |
| Replikate löschen | Verringern der Anzahl der Replikate in einem Shard | Verringern der Anzahl der Replikate in einem Shard Löschen eines Lesereplikats für Redis-Replikationsgruppen (Clustermodus deaktiviert) |

Erhöhen der Anzahl der Replikate in einem Shard

Sie können die Anzahl der Replikate in einem Redis-Replikationsgruppen-Shard (Cluster-Modus aktiviert) oder einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert) auf maximal fünf erhöhen. Verwenden Sie hierzu die AWS Management Console, den AWS CLI oder die ElastiCache-API.

Themen

- [Verwendung von AWS Management Console](#)
- [Verwendung von AWS CLI](#)
- [Verwenden der ElastiCache-API](#)

Verwendung von AWS Management Console

Das folgende Verfahren verwendet die Konsole, um die Anzahl der Replikate in einer Redis-Replikationsgruppe (Cluster-Modus aktiviert) zu erhöhen.

So erhöhen Sie die Anzahl von Replikaten in Redis-Shards

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis und danach 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, um ElastiCache die Wahl der Availability Zone für jedes neue Replikat zu überlassen, oder Availability Zones angeben, um selbst 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.

Verwendung von AWS CLI

Mithilfe des Befehls `increase-replica-count` mit den folgenden Parametern können Sie die Anzahl der Replikate in einem Redis-Shard erhöhen:

- `--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 Redis-Replikationsgruppen (Cluster-Modus deaktiviert), in denen es nur eine Knotengruppe oder Redis-Gruppe (Cluster-Modus aktiviert) gibt, oder wenn Sie möchten, dass alle Knotengruppen die gleiche 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 Redis-Gruppen (Cluster-Modus 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 Redis-Replikationsgruppen (Cluster-Modus deaktiviert) lautet die Shard-ID immer `0001`. Um die ID einer Redis-Knotengruppe (Cluster-Modus aktiviert) (Shards) zu finden, siehe [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 `--replica-configuration` weggelassen wird, wählt ElastiCache die Availability Zone für jedes neue Replikat 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 Anzahl gilt unabhängig davon, ob dies eine Redis (Cluster-Modus deaktiviert) mit einer einzelnen Knotengruppe oder eine Redis (Cluster-Modus aktiviert) mit mehreren Knotengruppen ist.

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 Redis-Replikationsgruppe (Cluster-Modus 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 der Replikate mithilfe des CLI finden Sie unter [increase-replica-count](#) in der Befehlszeilenreferenz für Amazon ElastiCache.

Verwenden der ElastiCache-API

Um die Anzahl der Replikate in einem Redis-Shard zu erhöhen, können Sie die Aktion `IncreaseReplicaCount` mit den folgenden Parametern verwenden:

- `ReplicationGroupId` – Erforderlich. Identifiziert, in welcher Replikationsgruppe Sie die Anzahl von Replikaten erhöhen möchten.
- `ApplyImmediately` – Erforderlich. Gibt an, ob die Replikatanzahl 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 Redis-Replikationsgruppen (Cluster-Modus deaktiviert), in denen es nur eine Knotengruppe gibt, oder für Redis-Gruppen (Cluster-Modus aktiviert), in denen alle Knotengruppen die gleiche 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 Redis-Gruppen (Cluster-Modus 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. Bei Redis-Replikationsgruppen (Cluster-Modus deaktiviert) ist die ID der Knotengruppe (Shard) immer 0001. Um die ID einer Redis-Knotengruppe (Cluster-Modus aktiviert) (Shards) zu finden, siehe [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 die Availability Zone für jedes neue Replikat 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 Anzahl gilt unabhängig davon, ob dies eine Redis (Cluster-Modus deaktiviert) mit einer einzelnen Knotengruppe oder eine Redis (Cluster-Modus aktiviert) mit mehreren Knotengruppen ist.

```
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 Redis-Replikationsgruppe (Cluster-Modus 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 der Replikate mithilfe der API finden Sie unter [IncreaseReplicaCount](#) in der Amazon-ElastiCache-API-Referenz.

Verringern der Anzahl der Replikate in einem Shard

Sie können die Anzahl der Replikate in einem Shard für Redis (Cluster-Modus aktiviert) oder in einer Replikationsgruppe für Redis (Cluster-Modus deaktiviert) verringern:

- Für Redis (Cluster-Modus deaktiviert) können Sie die Anzahl der Replikate auf eins verringern, wenn Multi-AZ aktiviert ist, und auf null, wenn es nicht aktiviert ist.
- Für Redis (Cluster-Modus 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 die AWS Management Console, die AWS CLI oder die ElastiCache API verwenden, um die Anzahl der Replikate in einer Knotengruppe (Shard) oder Replikationsgruppe zu verringern.

Themen

- [Mit dem AWS Management Console](#)
- [Unter Verwendung des AWS CLI](#)
- [Verwendung der API ElastiCache](#)

Mit dem AWS Management Console

Das folgende Verfahren verwendet die Konsole, um die Anzahl der Replikate in einer Redis-Replikationsgruppe (Cluster-Modus aktiviert) zu verringern.

So verringern Sie die Anzahl der Replikate in einem Redis-Shard

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 Redis und danach den Namen der Replikationsgruppe aus, 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 abzuberechnen.

Important

- Wenn Sie die zu löschenden Replikatknoten nicht angeben, wählt ElastiCache for Redis automatisch die zu löschenden Replikatknoten aus. Dabei versucht Redis, ElastiCache 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 Redis-Shard zu verringern, verwenden Sie den Befehl `decrease-replica-count` 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 der Knoten-IDs zur Angabe der Replikatsknoten, die entfernt werden sollen.

- `--replica-configuration` Optional. Ermöglicht Ihnen, die Anzahl der Replikate und Availability Zones für jede Knotengruppe separat festzulegen. Verwenden Sie diesen Parameter für Redis-Gruppen (Cluster-Modus 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 Redis-Replikationsgruppen (Cluster-Modus deaktiviert) lautet die Shard-ID immer `0001`. Um die ID einer Redis-Knotengruppe (Cluster-Modus aktiviert) (Shards) zu finden, siehe [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 Redis ElastiCache für jedes der neuen Replikate die Availability Zone 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 Anzahl gilt unabhängig davon, ob dies eine Redis (Cluster-Modus deaktiviert) mit einer einzelnen Knotengruppe oder eine Redis (Cluster-Modus aktiviert) mit mehreren Knotengruppen ist.

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 Redis-Replikationsgruppe (Cluster-Modus aktiviert). Wenn optional ein Wert für `PreferredAvailabilityZones` angegeben wird, muss die Anzahl der aufgelisteten `AvailabilityZones` 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 der CLI finden Sie unter [decrease-replica-count](#) in der Amazon-Befehlszeilenreferenz. ElastiCache

Verwendung der API ElastiCache

Um die Anzahl der Replikate in einem Redis-Shard zu verringern, können Sie die Aktion `DecreaseReplicaCount` mit den folgenden Parametern verwenden:

- `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 der Knoten-IDs zur Angabe der Replikatsknoten, die entfernt werden sollen.

- **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 Redis-Gruppen (Cluster-Modus 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 Redis-Replikationsgruppen (Cluster-Modus deaktiviert) lautet die Knotengruppen-ID immer 0001. Um die ID einer Redis-Knotengruppe (Cluster-Modus aktiviert) (Shards) zu finden, siehe [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 Redis ElastiCache für jedes der neuen Replikate die Availability Zone 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 Anzahl gilt unabhängig davon, ob dies eine Redis (Cluster-Modus deaktiviert) mit einer einzelnen Knotengruppe oder eine Redis (Cluster-Modus aktiviert) mit mehreren Knotengruppen ist.

```
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 Redis-Replikationsgruppe (Cluster-Modus 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, die die API verwenden, finden Sie unter [DecreaseReplicaAnzahl](#) in der Amazon ElastiCache API-Referenz.

Hinzufügen eines Lesereplikats für Redis-Replikationsgruppen (Clustermodus deaktiviert)

Die Informationen im folgenden Thema gelten nur für Redis-Replikationsgruppen (Cluster-Modus 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 ein Lesereplikat zu einem Redis-Cluster (Cluster-Modus deaktiviert) hinzufügen können.

Eine Redis (Cluster-Modus deaktiviert) -Replikationsgruppe kann maximal fünf Redis-Replikationsgruppen 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 Redis-Replikationsgruppe (Cluster-Modus aktiviert) finden Sie im Folgenden:

- [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#)

- [Erhöhen der Anzahl der Replikate in einem Shard](#)

Sie können ein Redis-Cluster (Cluster-Modus deaktiviert) mithilfe der ElastiCache Konsole, der AWS CLI oder die ElastiCache API.

Verwandte Themen

- [Knoten zu einem Cluster hinzufügen](#)
- [Hinzufügen einer Read Replica zu einer Replikationsgruppe \(AWS CLI\)](#)
- [Hinzufügen eines Read Replica zu einer Replikationsgruppe mithilfe der API](#)

Hinzufügen einer Read Replica zu einer Replikationsgruppe (AWS CLI)

Um ein Lesereplikat zu einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert) hinzuzufügen, geben Sie mittels des `create-cache-cluster`-Befehls AWS CLI mit dem Parameter `--replication-group-id` an, welche Replikationsgruppe zum 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 eines Read Replica mithilfe der CLI finden Sie unter [create-cache-cluster](#) im Befehlszeilenreferenz von Amazon ElastiCache.

Hinzufügen eines Read Replica zu einer Replikationsgruppe mithilfe der API

Um ein Lesereplikat zu einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert) hinzuzufügen, verwenden Sie die ElastiCache-Operation `CreateCacheCluster` mit dem Parameter `ReplicationGroupId` zur Angabe der Replikationsgruppe, zu der 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 eines Read Replica mithilfe der API finden Sie unter [CreateCacheCluster](#) in der Amazon-ElastiCache-API-Referenz.

Löschen eines Lesereplikats für Redis-Replikationsgruppen (Clustermodus deaktiviert)

Die Informationen im folgenden Thema gelten nur für Redis-Replikationsgruppen (Cluster-Modus deaktiviert).

Mit dem sich ändernden Datenverkehr durch Lesezugriffe für eine Redis-Replikationsgruppe, empfiehlt es sich möglicherweise, Read Replicas hinzuzufügen oder zu entfernen. Das Entfernen eines Knotens aus einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert) ist dasselbe wie das Löschen eines Clusters, allerdings gibt es 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 [Redis \(Cluster-Modus deaktiviert\) -Replikationsgruppen](#).

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 Redis-Replikationsgruppe (Cluster-Modus deaktiviert) mithilfe der ElastiCache Konsole, der AWS CLI für ElastiCache oder die ElastiCache API.

Anleitungen zum Löschen eines Clusters aus einer Redis-Replikationsgruppe finden Sie in:

- [Verwendung von AWS Management Console](#)
- [Verwendung der AWS CLI](#)
- [Verwenden der ElastiCache-API](#)
- [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#)
- [Verringern der Anzahl der Replikate in einem Shard](#)

Redis (Cluster-Modus deaktiviert) -Replikationsgruppen

Die Informationen im folgenden Thema gelten nur für Redis-Replikationsgruppen (Cluster-Modus deaktiviert).

Sie können eine Redis-Lesereplik (Clustermodus deaktiviert) mithilfe der AWS Management Console, AWS CLI, oder der API zur primären Replikation ElastiCache heraufstufen. Sie können ein Lesereplikat nicht zum primären Replikat erheben, wenn Multi-AZ mit automatischem Failover auf der Redis-Replikationsgruppe (Cluster-Modus deaktiviert) aktiviert ist. Um ein Redis-Replikat (Cluster-Modus deaktiviert) in einer Multi-AZ-fähigen Replikationsgruppe zum primären Replikat zu ernennen, gehen Sie wie folgt vor:

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, die Redis 2.6.13 oder früher ausführen, nicht verfügbar.

Mit dem 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 hochstufen möchten, Mitglied einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert) ist, in der Multi-AZ aktiviert ist, ändern Sie die Replikationsgruppe, um Multi-AZ zu deaktivieren, bevor Sie fortfahren. Weitere Informationen finden Sie unter [Ändern einer Replikationsgruppe](#).
3. Wählen Sie Redis. Wählen Sie dann aus der Liste der Cluster 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.

4. 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.
5. Führen Sie im Dialogfeld Promote Read Replica (Read Replica heraufstufen) die folgenden Schritte aus:
 - a. 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.
 - b. Wählen Sie Promote, um die Read Replica heraufzustufen, oder Cancel, um die Operation abubrechen.
6. 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 [modify-replication-group](#) in der Amazon ElastiCache Command Line Reference.

Verwenden der 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 [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

Verwaltung der Wartung

Jeder Cluster und jede Replikationsgruppe hat ein wöchentliches Wartungsfenster, in dem Systemänderungen zugewiesen werden. Wenn Sie beim Erstellen oder Ändern eines Clusters oder

einer Replikationsgruppe kein bevorzugtes Wartungsfenster angeben, weist ElastiCache an einem zufällig ausgewählten Wochentag ein 60-minütiges Wartungsfenster innerhalb des Wartungsfensters Ihrer Region 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 UTC |
| ap-northeast-3 | Region Asien-Pazifik (Osaka) | 12:00 - 20:00 UTC |
| ap-southeast-3 | Region Asien-Pazifik (Jakarta) | 14:00 - 22:00 UHR UTC |
| ap-south-1 | Region Asien-Pazifik (Mumbai) | 17:30 - 1:30 UHR UTC |
| ap-southeast-1 | Region Asien-Pazifik (Singapur) | 14:00 - 22:00 UHR UTC |
| cn-north-1 | Region China (Peking) | 14:00 - 22:00 UHR UTC |
| cn-northwest-1 | Region China (Ningxia) | 14:00 - 22:00 UHR UTC |
| ap-east-1 | Region Asien-Pazifik (Hongkong) | 13:00 - 21:00 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 UTC |
| eu-central-1 | Region Europa (Frankfurt) | 23:00 - 07:00 UTC |
| eu-west-1 | Region Europa (Irland) | 22:00 bis 06:00 Uhr UTC |
| eu-west-2 | Region Europa (London) | 23:00 - 07:00 UTC |

| Regionscode | Name der Region | Regionale Wartungsfenster |
|---------------|--------------------------------|---------------------------|
| me-south-1 | Region Naher Osten (Bahrain) | 13:00 - 21:00 UTC |
| me-central-1 | Region „Naher Osten (VAE)“ | 13:00 - 21:00 UTC |
| eu-south-1 | Region Europa (Mailand) | 21:00 - 05:00 UTC |
| sa-east-1 | Region Südamerika (São Paulo) | 01:00 - 09:00 UHR UTC |
| us-east-1 | Region USA Ost (Nord-Virginia) | 03:00 - 11:00 UHR UTC |
| us-east-2 | Region USA Ost (Ohio) | 04:00 - 12:00 UHR UTC |
| us-gov-west-1 | Region AWS GovCloud (US) | 06:00 bis 14:00 Uhr UTC |
| us-west-1 | Region US West (N. California) | 06:00 - 14:00 UHR UTC |
| us-west-2 | Region USA West (Oregon) | 06:00 bis 14:00 Uhr UTC |

Ändern des Wartungsfensters für den Cluster oder die Replikationsgruppe

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 über die AWS Management Console sofort anwenden möchten, wählen Sie das Feld **Apply immediately** (Sofort anwenden) aus. Andernfalls werden diese Änderungen während des nächsten geplanten Wartungsfensters angewendet. Wenn Sie die API verwenden möchten, beachten Sie die Informationen unter [modify-replication-group](#) oder [modify-cache-cluster](#).

Weitere Informationen

Informationen zu Ihrem Wartungsfenster und dem Austausch von Knoten finden Sie unter:

- [ElastiCache-Wartung](#) – Häufig gestellte Fragen zu Wartung und Knotenaustausch
- [Ersetzen von Knoten](#) – Verwalten des Knotenaustauschs
- [Ändern einer Replikationsgruppe](#) – Ändern des Wartungsfensters einer Replikationsgruppe

Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen

Amazon ElastiCache steuert die Laufzeiteigenschaften Ihrer Knoten und Cluster mithilfe von Parametern. In der Regel enthalten neuere Engine-Versionen zusätzliche Parameter zur Unterstützung der neueren Funktionalität. Parametertabellen finden Sie unter [Redis-spezifische Parameter](#).

Wie zu erwarten, werden einige Parameterwerte, wie z. B. `maxmemory` durch die Engine und den Knotentyp bestimmt. Eine Tabelle dieser Parameterwerte nach Knotentyp finden Sie unter [Spezifische Parameter für den Redis-Knotentyp](#).

Themen

- [Parameterverwaltung](#)
- [Stufen der Cache-Parametergruppen](#)
- [Erstellen einer Parametergruppe](#)
- [Auflisten von Parametergruppen nach Namen](#)
- [Auflisten der Werte einer Parametergruppe](#)
- [Modifizieren einer Parametergruppe](#)
- [Löschen einer Parametergruppe](#)
- [Memcached-spezifische Parameter](#)
- [Redis-spezifische Parameter](#)

Parameterverwaltung

Parameter werden zur leichteren Parameterverwaltung in benannte Gruppen zusammengruppiert. 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.
- Die Familie der Parametergruppe und der Cluster, dem Sie sie zuordnen, müssen kompatibel sein. Wenn Ihr Cluster z. B. Redis Version 3.2.10 ausführt, können Sie nur Parametergruppen, Standard oder benutzerdefiniert, von der Redis 3.2-Familie verwenden.
- 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 die Parameter eines Clusters ändern, wird die Änderung entweder sofort oder, mit den folgenden Ausnahmen, nachdem die Cluster-Knoten neu gestartet wurden, 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 [Redis-spezifische Parameter](#) können Sie bestimmen, wann eine bestimmte Parameteränderung übernommen wird.

Weitere Informationen finden Sie unter [Knoten neu starten](#).

Redis-Parameteränderungen (Cluster Mode Aktiviert)

Führen Sie nachstehende Schritte aus, wenn Sie Änderungen an den folgenden Parametern in einem Redis-Cluster (Clustermodus aktiviert) vornehmen.

- activerehashing
- databases

1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
2. Löschen Sie den Redis-Cluster (Cluster-Modus aktiviert). 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.

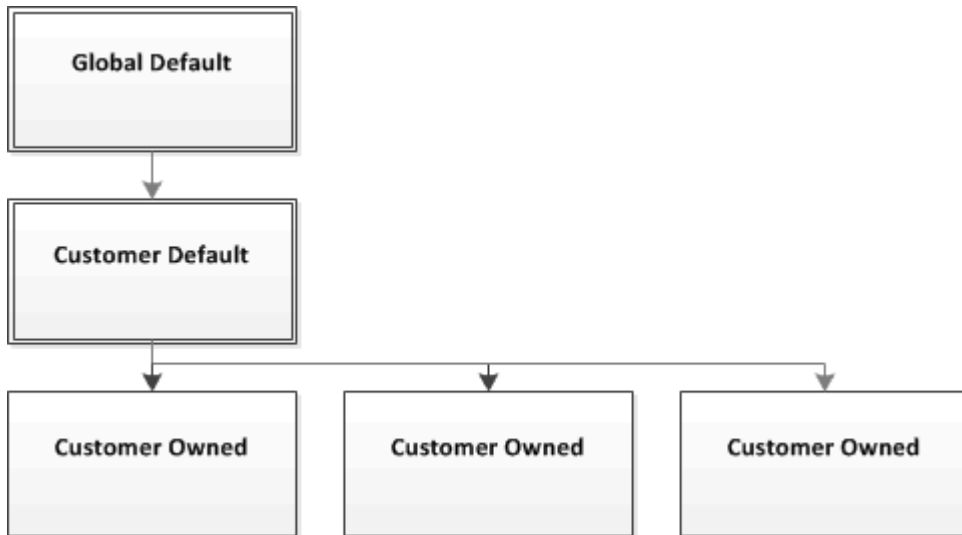
- Sie können Parametergruppen globalen Redis-Datenspeichern zuordnen. Globale Datenspeicher sind eine Sammlung von einem oder mehreren Clustern, die sich über mehrere AWS-Regionen erstrecken. In diesem Fall wird die Parametergruppe von allen Clustern gemeinsam genutzt, die den globalen Datenspeicher 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:

- Auf der ElastiCache-Konsole auf der Seite Parametergruppen, das Ja/nein im Global-Attribut
- Die Ja/nein-IsGlobalEigenschaft der API-Operation [CacheParameterGroup](#)

Stufen der Cache-Parametergruppen

Amazon ElastiCache hat drei Ebenen von Cache-Parametergruppen, wie nachfolgend gezeigt.



Stufen der Parametergruppen von Amazon ElastiCache

Global Default

Die oberste Stammparametergruppe für alle Amazon-ElastiCache-Kunden in der Region.

Die globale Standard-Cache-Parametergruppe:

- Sie ist für ElastiCache reserviert und für den Kunden nicht verfügbar.

Customer Default

Eine Kopie der Cache-Parametergruppe „Global Default“, die für die Verwendung durch den Kunden erstellt wird.

Die Cache-Parametergruppe „Customer Default“:

- Wird von ElastiCache erstellt und ist dessen Eigentum.
- 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 finden Sie unter [Redis-spezifische Parameter](#).

Erstellen einer Parametergruppe

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, dem AWS CLI oder der ElastiCache-API erstellen.

Erstellen einer Parametergruppe (Konsole)

Mit dem folgenden Verfahren wird eine Parametergruppe mithilfe der ElastiCache-Konsole erstellt.

So erstellen Sie eine Parametergruppe mithilfe der ElastiCache-Konsole

1. Melden Sie sich bei AWS Management Console an 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. redis3.2, definiert die tatsächlichen Parameter in Ihrer Parametergruppe und deren anfänglichen Werte. 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:

- Er muss mit einem ASCII-Buchstaben beginnen.
 - Er 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 [Modifizieren einer Parametergruppe](#).

Erstellen einer Parametergruppe (AWS CLI)

Um mithilfe der AWS CLI eine Parametergruppe zu erstellen, verwenden Sie den Befehl `create-cache-parameter-group` mit diesen Parametern.

- `--cache-parameter-group-name` Der Name der Parametergruppe.

Namenseinschränkungen für die Parametergruppe:

- Er muss mit einem ASCII-Buchstaben beginnen.
 - Er 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 anhand der redis2.8-Familie als Vorlage eine Parametergruppe namens myRed28 erstellt.

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 [Modifizieren einer Parametergruppe](#).

Weitere Informationen finden Sie unter [create-cache-parameter-group](#).

Erstellen einer Parametergruppe (ElastiCache API)

Um mithilfe der ElastiCache-API eine Parametergruppe zu erstellen, verwenden Sie die Aktion `CreateCacheParameterGroup` mit diesen Parametern.

- `ParameterGroupName` Der Name der Parametergruppe.

Namenseinschränkungen für die Parametergruppe:

- Er muss mit einem ASCII-Buchstaben beginnen.
- Er 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. Zum Beispiel `redis2.8`.
- `Description` Eine vom Benutzer eingegebene Beschreibung der Parametergruppe.

Example

Im folgenden Beispiel wird anhand der `redis2.8`-Familie als Vorlage eine Parametergruppe namens `myRed28` erstellt.

```
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 [Modifizieren einer Parametergruppe](#).

Weitere Informationen finden Sie unter [CreateCacheParameterGroup](#).

Auflisten von Parametergruppen nach Namen

Sie können die Parametergruppen mithilfe der ElastiCache Konsole, der AWS CLI oder die ElastiCache API.

Auflisten von Parametergruppen nach Namen (Konsole)

Mit dem folgenden Verfahren wird eine Liste der Parametergruppen mithilfe der ElastiCache-Konsole angezeigt.

So listen Sie Parametergruppen mittels der ElastiCache-Konsole auf

1. Melden Sie sich bei AWS Management Console an 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.

Auflisten von Parametergruppen nach Namen (AWS CLI)

Verwenden Sie zum Generieren einer Liste von Parametergruppen mithilfe der AWS CLI den Befehl `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 Fällen werden Name, Familie und Beschreibung der Parametergruppe aufgelistet.

Example

Der folgenden 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

Im folgenden Beispiel-Code wird die Parametergruppe MyRed56 für Parametergruppen aufgeführt, die ab Redis Engine-Version 5.0.6 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 folgendermaßen aus und listet den Namen, die Familie „isGlobal“ und die Beschreibung der 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 und im Fall von redis5.6, ob die Parametergruppe Teil eines globalen Datenspeichers (IsGlobal) ist, auf.

```
{
  "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](#).

Auflisten von Parametergruppen nach Namen (ElastiCache API)

Verwenden Sie zum Generieren einer Liste von Parametergruppen mithilfe der ElastiCache-API die Aktion `DescribeCacheParameterGroups`. 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 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 dieser Aktion sieht ungefähr so aus und listet für jede Parametergruppe den Namen, die Familie, die Beschreibung und im Fall von redis5.6, ob die Parametergruppe zu einem globalen Datenspeicher (`IsGlobal`) gehört, 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 folgenden Beispielcode listet die Parametergruppe myRed28 auf.

```

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 Beispiel-Code 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 dieser Aktion sieht ungefähr so aus und listet den Namen, die Familie, die Beschreibung, und ob die Parametergruppe Teil eines globalen Datenspeichers (IsGlobal) ist, auf.

```

<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](#).

Auflisten der Werte einer Parametergruppe

Sie können die Parameter und deren Werte für eine Parametergruppe mithilfe der ElastiCache-Konsole, dem AWS CLI oder der ElastiCache-API auflisten.

Auflisten der Werte einer Parametergruppe (Konsole)

Mit dem folgenden Verfahren wird eine Liste der Parameter und deren Werte für eine Parametergruppe mithilfe der ElastiCache-Konsole aufgelistet.

So listen Sie die Parameter und deren Werte für eine Parametergruppe mithilfe der ElastiCache-Konsole auf

1. Melden Sie sich bei AWS Management Console an 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 und deren Werte einer Parametergruppe mithilfe der AWS CLI aufzulisten, verwenden Sie den Befehl `describe-cache-parameters`.

Example

Der folgende Beispielcode listet alle Parameter und deren Werte für die Parametergruppe `myRedis28` 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)

Um die Parameter und deren Werte einer Parametergruppe mithilfe der ElastiCache-API aufzulisten, verwenden Sie die Aktion DescribeCacheParameters.

Example

Der folgende Beispielcode listet alle Parameter für die Parametergruppe myRed28 auf.

```
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>
```

...output omitted...

```
</CacheClusterClassSpecificParameters>
</DescribeCacheParametersResult>
<ResponseMetadata>
  <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParametersResponse>
```

Weitere Informationen finden Sie unter [DescribeCacheParameters](#).

Modifizieren einer 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 über den Zeitpunkt, an dem die Änderung eines Parameterwertes von einer Parametergruppe übernommen wird, finden Sie unter [Redis-spezifische Parameter](#).

Ändern einer Parametergruppe (Konsole)

Im folgenden Verfahren wird gezeigt, wie der Wert des Parameters `cluster-enabled` in der ElastiCache-Konsole bearbeitet wird. Dieses Verfahren ist bei der Bearbeitung des Wertes aller Parameter gleich.

So ändern Sie den Wert eines Parameters über die ElastiCache-Konsole

1. Melden Sie sich bei AWS Management Console an 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. Wählen Sie Save Changes.
6. Den Namen des Parameters, den Sie in einem dieser Themen bearbeitet haben, finden Sie unter [Redis-spezifische Parameter](#). Wenn Sie einen Redis-Cluster (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](#).

Redis-Parameteränderungen (Cluster Mode Aktiviert)

Führen Sie nachstehende Schritte aus, wenn Sie Änderungen an den folgenden Parametern in einem Redis-Cluster (Clustermodus aktiviert) vornehmen.

- activerehashing
- databases

1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
2. Löschen Sie den Redis-Cluster (Cluster-Modus aktiviert). 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 mithilfe des AWS CLI zu ändern, verwenden Sie den Befehl `modify-cache-parameter-group`.

Example

Den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, finden Sie unter [Redis-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `reserved-memory-percent` und `cluster-aktiviert` auf der Parametergruppe `myredis32-on-30`. Wir legen `reserved-memory-percent` auf `30` (30 Prozent) und `cluster-enabled` auf `yes` fest, damit die Parametergruppe mit Redis-Clustern (Cluster-Modus 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 [Redis-spezifische Parameter](#).

Wenn Sie einen Redis-Cluster (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](#).

Redis-Parameteränderungen (Cluster Mode Aktiviert)

Führen Sie nachstehende Schritte aus, wenn Sie Änderungen an den folgenden Parametern in einem Redis-Cluster (Clustermodus aktiviert) vornehmen.

- activerehashing
 - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Redis-Cluster (Cluster-Modus aktiviert). 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)

Um die Parameterwerte einer Parametergruppe mithilfe der ElastiCache-API zu ändern, verwenden Sie die Aktion `ModifyCacheParameterGroup`.

Example

Den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, finden Sie unter [Redis-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `reserved-memory-percent` und `cluster-aktiviert` auf der Parametergruppe `myredis32-on-30`. Wir legen `reserved-memory-percent` auf `30` (30 Prozent) und `cluster-enabled` auf `yes` fest, damit die Parametergruppe mit Redis-Clustern (Cluster-Modus 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 Redis-Cluster (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](#).

Redis-Parameteränderungen (Cluster Mode Aktiviert)

Führen Sie nachstehende Schritte aus, wenn Sie Änderungen an den folgenden Parametern in einem Redis-Cluster (Clustermodus aktiviert) vornehmen.

- activerehashing
 - databases
1. Erstellen Sie eine manuelle Sicherung des Clusters. Siehe [Erstellen manueller Backups](#).
 2. Löschen Sie den Redis-Cluster (Cluster-Modus aktiviert). Siehe [Löschen eines Clusters](#).
 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 Parametergruppe

Sie können eine benutzerdefinierte Parametergruppe mithilfe der ElastiCache-Konsole, dem AWS CLI oder der ElastiCache-API löschen.

Parametergruppen, die Clustern zugeordnet sind, können nicht gelöscht werden. Standard-Parametergruppen können ebenfalls nicht gelöscht werden.

Löschen einer Parametergruppe (Konsole)

Mit dem folgenden Verfahren wird eine Parametergruppe mithilfe der ElastiCache-Konsole gelöscht.

So löschen Sie eine Parametergruppe mithilfe der ElastiCache-Konsole

1. Melden Sie sich bei AWS Management Console an 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 Delete (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 mithilfe der AWS CLI zu löschen, verwenden Sie den Befehl `delete-cache-parameter-group`. Der durch `--cache-parameter-group-name` angegebene Parametergruppe dürfen eine Cluster zugeordnet sein und es darf sich nicht um eine Standard-Parametergruppe handeln, damit sie gelöscht wird.

Der folgenden Beispielcode löscht die Parametergruppe `myMem14`.

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)

Um eine Parametergruppe mithilfe der ElastiCache-API zu löschen, verwenden Sie die Aktion `DeleteCacheParameterGroup`. 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

Der folgenden Beispielcode löscht die Parametergruppe `myRed28`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteCacheParameterGroup  
&CacheParameterGroupName=myRed28  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie unter [DeleteCacheParameterGroup](#).

Memcached-spezifische Parameter

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 [Erstellen einer Parametergruppe](#).

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 |
|-----------|--|--|
| no_modern | <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 zur Deaktivierung von <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> , <code>maxconns_fast</code> Befehlen. <code>No modern</code> setzt auch <code>hash_algorithm</code> auf <code>jenkins</code> und erlaubt das Inlining von ASCII VALUE. Anwendbar auf Memcached 1.5 und höher</p> <p>Um zu „modern“ zurückzukehren, müssen Sie diesen Parameter deaktivieren und neu starten, wodurch <code>slab_reassign</code> , <code>slab_automove</code> , <code>lru_crawler</code> , <code>lru_maintainer</code> und <code>maxconns_fast</code> automatisch aktiviert werden.</p> <div data-bbox="1008 1171 1511 1885"><p> Note</p><p>Der Standardkonfigurationswert für diesen Parameter wurde am 20. August 2021 von 0 in 1 geändert. Nach dem 20. August 2021 wird von neuen Elasticache-Benutzern für jede Region automatisch der aktualisierte Standardwert übernommen. Vor dem 20. August 2021 bestehende ElastiCache-</p></div> |

| Name | Details | Beschreibung |
|--------------------------------|---|--|
| | | Benutzer in den Regionen müssen ihre benutzerdefinierten Parametergruppen manuell ändern, um diese neue Änderung zu übernehmen. |
| <code>inline_ascii_resp</code> | Standard: 0 Typ: Boolesch Veränderbar: Ja Zulässige Werte: 0, 1 Änderungen werden wirksam: Beim Start | Speichert Zahlen der VALUE-Antwort innerhalb eines Elements mit bis zu 24 Bytes. Geringe Verlangsamung für ASCII-Sätze <code>get</code> und <code>fastset</code> . |

Für Memcached 1.5.10 werden die folgenden Parameter entfernt.

| Name | Details | Beschreibung |
|---|---|---|
| <code>expirezero_does_no_t_evict</code> | Standard: 0 Typ: Boolesch Veränderbar: Ja Zulässige Werte: 0, 1 Änderungen werden wirksam: Beim Start | Wird in dieser Version nicht länger unterstützt |
| <code>modern</code> | Standard: 1 Typ: Boolesch | Wird in dieser Version nicht länger unterstützt Ab dieser Version ist |

| Name | Details | Beschreibung |
|------|--|--|
| | <p>Anpassbar: Ja (erfordert Neustart, wenn auf <code>no_modern</code> gesetzt)</p> <p>Zulässige Werte: 0, 1</p> <p>Änderungen werden wirksam: Beim Start</p> | <p><code>no-modern</code> standardmäßig bei jedem Start oder Neustart aktiviert.</p> |

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> | <p>Standard: aktiviert</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p> | <p>Ein Alias für mehrere Funktionen. Das Aktivieren von <code>modern</code> ist gleichbedeutend mit dem Einschalten der folgenden Befehle und der Verwendung eines murmur3-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>.</p> |
| <code>watch</code> | <p>Standard: aktiviert</p> | |

| Name | Details | Beschreibung |
|---------------------------|---|---|
| | <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> | <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.</p> |
| <code>idle_timeout</code> | <p>Standard: 0 (deaktiviert)</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Beim Start</p> | <p>Die minimale Anzahl von Sekunden, die sich ein Client im Leerlauf befinden kann, bevor er zum Schließen aufgefordert wird. Wertebereich: 0 bis 86400.</p> |
| <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> |

| Name | Details | Beschreibung |
|----------------------------------|--|--|
| <code>watcher_logbuf_size</code> | Standard: 256 (KB) Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Beim Start | 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. |
| <code>worker_logbuf_size</code> | Standard: 64 (KB) Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Beim Start | 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. |
| <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. |



| Name | Details | Beschreibung |
|---|--|--|
| <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. Werte: 0, 1 (Benutzer kann einen <code>flush_all</code> ausführen, wenn der Wert 0 ist). |
| <code>hash_algorithm</code> | Standard: jenkins Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Beim Start | Der zu verwendende Hash-Algorithmus. Zulässige Werte: <code>murmur3</code> und <code>jenkins</code> . |

| 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> Note</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> Note</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 Modifizieren</p> </div> |

| Name | Details | Beschreibung |
|--|---|--|
| | | <p>ren einer Parametergruppe.</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 LRUs verlagert, wenn Kapazitäten erreicht werden. 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> Warning</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-Hash-Tabelle, 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 eingestel |

| Name | Beschreibung |
|---------------|---|
| | It wird, sendet ElastiCache einen Fehler zum Client und schließt umgehend die Verbindung. |
| slab_automove | Korrigiert den Slab-AutoMove-Algorithmus: Wenn dieser Parameter auf 0 (Null) eingestellt ist, ist der AutoMove-Algorithmus deaktiviert. Wenn er auf 1 eingestellt ist, geht ElastiCache beim automatischen Verschieben von Slabs langsam und zurückhaltend vor. Wenn er auf 2 eingestellt ist, verschiebt ElastiCache Slabs aggressiv bei jeder Bereinigung. (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 |
|--|---|--|
| | <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Nach dem Neustart</p> | <p>Eine Anleitung zum Ändern des Wertes von <code>binding_protocol</code> finden Sie unter Modifizieren einer Parametergruppe.</p> |
| <code>cas_disabled</code> | <p>Standard: 0(false)</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Nach dem Neustart</p> | <p>Bei dem Wert 1 (true) werden die Operationen <code>check</code> und <code>set</code> (CAS) deaktiviert und gespeicherte Elemente verbrauchen 8 Bytes weniger als bei aktiviertem CAS.</p> |
| <code>chunk_size</code> | <p>Standard: 48</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Nach dem Neustart</p> | <p>Die Mindestmenge, in Byte, von Speicher, die für den Schlüssel, Wert und die Flags des kleinsten Elements zuzuordnen sind.</p> |
| <code>chunk_size_growth_factor</code> | <p>Standard: 1.25</p> <p>Typ: Gleitkommazahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Nach dem Neustart</p> | <p>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.</p> |
| <code>error_on_memory_exhausted</code> | <p>Standard: 0(false)</p> <p>Typ: Boolesch</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Nach dem Neustart</p> | <p>Bei dem Wert 1 (true) gibt Memcached einen Fehler zurück, wenn kein Speicher zum Speichern der Elemente mehr verfügbar ist, anstatt Elemente zu bereinigen.</p> |

| Name | Details | Beschreibung |
|------------------------------|--|---|
| large_memory_pages | Standard: 0(false) Typ: Boolesch Veränderbar: Nein | Bei dem Wert 1 (true) versucht ElastiCache, größere Speicherseiten zu verwenden. |
| lock_down_paged_memory | Standard: 0(false) Typ: Boolesch Veränderbar: Nein | Bei dem Wert 1 (true) sperrt ElastiCache den Zugriff auf den gesamten ausgelagerten Speicher. |
| 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 | Bei dem Wert 1 (true) maximiert ElastiCache das Limit der Kerndatei. |

| 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 Overhead von Memcached-Verbindungen . |
| <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 Standard-Parametergruppe nutzt, legt ElastiCache als Wert für `memcached_connections_overhead` 13 MB fest.
- 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 |
| cache.r6g.2xlarge | 54084 | 8 |
| cache.r6g.4xlarge | 108347 | 16 |
| cache.r6g.8xlarge | 214577 | 32 |
| cache.r6g.12xlarge | 325400 | 48 |
| cache.r6g.16xlarge | 429154 | 64 |
| cache.c7gn.large | 3164 | 2 |
| cache.c7gn.xlarge | 6537 | 4 |
| cache.c7gn.2xlarge | 13248 | 8 |
| cache.c7gn.4xlarge | 26671 | 16 |

| 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 (Amazon VPC) erstellt.

Redis-spezifische Parameter

Wenn Sie für Ihren Redis-Cluster keine Parametergruppe angeben, wird eine für Ihre Engine-Version geeignete Standard-Parametergruppe verwendet. 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 [Erstellen einer Parametergruppe](#).

Themen

- [Parameteränderungen bei Redis 7](#)
- [Parameteränderungen von Redis 6.x](#)
- [Redis 5.0.3-Parameteränderungen](#)
- [Redis 5.0.0-Parameteränderungen](#)
- [Redis 4.0.10-Parameteränderungen](#)
- [Redis 3.2.10-Parameteränderungen](#)
- [Redis 3.2.6-Parameteränderungen](#)
- [Redis 3.2.4-Parameteränderungen](#)
- [In Redis 2.8.24 \(Enhanced\) hinzugefügte Parameter](#)
- [In Redis 2.8.23 \(Enhanced\) hinzugefügte Parameter](#)
- [In Redis 2.8.22 \(Enhanced\) hinzugefügte Parameter](#)
- [In Redis 2.8.21 hinzugefügte Parameter](#)
- [In Redis 2.8.19 hinzugefügte Parameter](#)
- [In Redis 2.8.6 hinzugefügte Parameter](#)
- [Parameter von Redis 2.6.13](#)
- [Spezifische Parameter für den Redis-Knotentyp](#)

Parameteränderungen bei Redis 7

Familie der Parametergruppe: redis7

Die Standardparametergruppen von Redis 7 lauten wie folgt:

- `default.redis7` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus deaktiviert) und Replikationsgruppen.

- `default.redis7.cluster.on` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus aktiviert) und Replikationsgruppen.

Die in Redis 7 hinzugefügten Parameter lauten wie folgt.

| Name | Details | Beschreibung |
|--|--|---|
| <code>cluster-allow-pubsubshard-when-down</code> | <p>Gültige Werte: <code>yes</code>, <code>no</code></p> <p>Standard: <code>yes</code></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: <code>ip</code>, <code>tls-dynamic</code></p> <p>Standard: <code>tls-dynamic</code></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-Anforderungen 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 <code>tls-dynamic</code> gesetzt ist, gibt der Knoten einen Hostnamen bekannt, wenn er aktiviert <code>encryption-in-transit</code> ist, und andernfalls eine IP-Adresse.</p> |
| <code>latency-tracking</code> | <p>Gültige Werte: <code>yes</code>, <code>no</code></p> <p>Standard: <code>no</code></p> <p>Typ: Zeichenfolge</p> <p>Veränderbar: Ja</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</p> |

| Name | Details | Beschreibung |
|----------------------------------|--|--|
| | <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p> | <p>(Histogramme) über den LATENCY-Befehl exportiert.</p> |
| <p>hash-max-listpack-entries</p> | <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> |
| <p>hash-max-listpack-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>Der Schwellenwert der maximalen Anzahl von Hash-Einträgen, damit der Datensatz komprimiert werden kann.</p> |

| Name | Details | Beschreibung |
|--|--|---|
| <code>zset-max-listpack-entries</code> | <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> | Die maximale Anzahl von Einträgen zu sortierten Sätzen, damit der Datensatz komprimiert werden kann. |
| <code>zset-max-listpack-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> | Der Schwellenwert der maximalen Anzahl von Einträgen zu sortierten Sätzen, damit der Datensatz komprimiert werden kann. |

Die in Redis 7 geänderten Parameter lauten wie folgt.

| Name | Details | Beschreibung |
|-------------------------------|--|---------------------|
| <code>activeresharding</code> | <p>Anpassbar: no. In Redis 7 ist dieser Parameter standardmäßig ausgeblendet und aktiviert. Wenn Sie ihn deaktivieren möchten,</p> | Veränderbar war ja. |

| Name | Details | Beschreibung |
|------|---|--------------|
| | müssen Sie einen Support-Fall erstellen . | |

Die in Redis 7 entfernten Parameter lauten wie folgt.

| Name | Details | Beschreibung |
|--------------------------|--|--|
| hash-max-ziplist-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> | Für die Darstellung kleiner Hash-Kodierungen listpack statt ziplist verwenden |
| hash-max-ziplist-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> | Für die Darstellung kleiner Hash-Kodierungen listpack statt ziplist verwenden |
| zset-max-ziplist-entries | <p>Zulässige Werte: 0+</p> <p>Standard: 128</p> <p>Typ: Ganzzahl</p> | Verwenden Sie listpack statt ziplist für die Darstellung kleiner Hash-Kodierungen. |

| Name | Details | Beschreibung |
|-------------------------------------|---|--|
| | <p>Veränderbar: Ja</p> <p>Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster</p> | |
| <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. |

Parameteränderungen von Redis 6.x

Familie der Parametergruppe: `redis6.x`

Redis 6.x Standardparametergruppen lauten wie folgt:

- `default.redis6.x` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus deaktiviert) und Replikationsgruppen.

- `default.redis6.x.cluster.on` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus aktiviert) und Replikationsgruppen.

Note

In Redis-Engine Version 6.2, als die R6gd-Knotenfamilie zur Verwendung mit [Daten-Tiering](#) eingeführt wurde, werden nur die maxmemory-Richtlinien `noeviction`, `volatile-lru` und `allkeys-lru` mit R6gd-Knotentypen unterstützt.

Weitere Informationen finden Sie unter [ElastiCache for Redis Version 6.2 \(erweitert\)](#) und [ElastiCache for Redis Version 6.0 \(erweitert\)](#).

Die in Redis 6.x hinzugefügten Parameter sind wie folgt.

| Name | 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>Wann die Änderungen wirksam werden: Die bestehenden Redis-Benutzer, die dem Cluster zugeordnet sind, verfügen weiterhin über vorhandene Berechtigungen. Aktualisieren Sie entweder die Benutzer oder starten Sie den Cluster neu, um die bestehenden Redis-Benutzer zu aktualisieren.</p> | Standardberechtigungen für Pub/Sub-Kanäle für ACL-Benutzer, die in diesem Cluster bereitgestellt werden. |

| Name | Details | Beschreibung |
|---|---|--|
| <code>cluster-allow-reads-when-down</code> (added in 6.0) | Standard: no Typ: Zeichenfolge Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster | <p>Wenn diese Option auf yes gesetzt ist, verarbeitet eine Redis-Replikationsgruppe (Clustermodus aktiviert) auch dann Lesebefehle, wenn ein Knoten kein Quorum von Primärwerten erreichen kann.</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> |
| <code>tracking-table-max-keys</code> (added in 6.0) | Standard: 1.000.000 Typ: Zahl Veränderbar: Ja Änderungen treten in Kraft: unmittelbar auf allen Knoten im Cluster | <p>Um das clientseitige Caching zu unterstützen, unterstützt Redis das Nachverfolgen, 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> |

| Name | 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 ACL-Protokoll.</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 löscht Schlüssel, die ihre Zeit überschritten haben, um von zwei Mechanismen zu leben. 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 zum Ablaufen von Elementen im periodischen Auftrag verwendet.</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 ist eine höhere CPU und eine potenziell höhere Latenz. Wir empfehlen den Wert 1, es sei denn, Sie sehen eine hohe Speicherauslastung und können eine Erhöhung der CPU-Auslastung tolerieren.</p> |

| Name | 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. |

Die in Redis 6.x entfernten Parameter sind wie folgt.

| 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. |

Redis 5.0.3-Parameteränderungen

Familie der Parametergruppe: redis5.0

Standard-Parametergruppen in Redis 5.0

- `default.redis5.0` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus deaktiviert) und Replikationsgruppen.
- `default.redis5.0.cluster.on` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus aktiviert) und Replikationsgruppen.

In Redis 5.0.3 hinzugefügte Parameter

| Name | Details | Beschreibung |
|-----------------|--|--|
| rename-commands | <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 der umbenannten Redis-Befehle. Im Folgenden finden Sie eine eingeschränkte Liste der verfügbaren Befehle:</p> <p>APPEND AUTH BITCOUNT BITFIELD BITOP BITPOS BLPOP BRPOP BRPOPLUSH 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 LPOS LPUSH 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 RPOPLUSH RPUSH RPUSHX SADD SCARD SCRIPT SDIFF SDIFFSTORE SELECT SET SETBIT SETEX SETNX SETRANGE SINTER SINTERSTORE SISMEMBER SLOWLOG SMEMBERS SMOVE SORT SPOP</p> |

| Name | Details | Beschreibung |
|------|---------|--|
| | | SRANDMEMBER SREM STRLEN SUBSCRIBE SUNION SUNIONSTORE 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 for Redis Version 5.0.6 \(erweitert\)](#).

Redis 5.0.0-Parameteränderungen

Familie der Parametergruppe: redis5.0

Standard-Parametergruppen in Redis 5.0

- `default.redis5.0` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus deaktiviert) und Replikationsgruppen.
- `default.redis5.0.cluster.on` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus aktiviert) und Replikationsgruppen.

In Redis 5.0 hinzugefügte Parameter

| 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> | 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. |
| <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> | 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. |
| <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> | Maximale Anzahl an Set-/Hash-/Zset-/Listenfeldern, die vom Scan des Hauptwörterbuchs verarbeitet werden |

| Name | Details | Beschreibung |
|---------------------------------------|---|---|
| <code>lua-replicate-commands</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-ignore-maxmemory</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 |

Aufgrund des Feedbacks der Community hat Redis für die Engine-Version 5.0 mehrere Parameter umbenannt. Weitere Informationen finden Sie unter [What's New in Redis 5? \(Neuerungen für Redis 5\)](#). In der folgenden Tabelle sind die neuen Parameternamen und deren Entsprechung in früheren Versionen angegeben.

Umbenannte Parameter in Redis 5.0

| Name | Details | Beschreibung |
|---------------------------------|---|---|
| <code>replica-lazy-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> | Standard: Für Werte siehe Spezifische Parameter für den Redis-Knotentyp Typ: Ganzzahl Veränderbar: Nein Ehemaliger Name: <code>client-output-buffer-limit-slave-hard-limit</code> | Für Redis-Read Replicas: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Bytes erreicht, wird die Verbindung zum Client getrennt. |
| <code>client-output-buffer-limit-replica-soft-limit</code> | Standard: Für Werte siehe Spezifische Parameter für den Redis-Knotentyp Typ: Ganzzahl Veränderbar: Nein Ehemaliger Name: <code>client-output-buffer-limit-slave-soft-limit</code> | Für Redis-Read Replicas: 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-replica-soft-seconds</code> lang andauert. |
| <code>client-output-buffer-limit-replica-soft-seconds</code> | Standard: 60 Typ: Ganzzahl Veränderbar: Nein Ehemaliger Name: <code>client-output-buffer-limit-slave-soft-seconds</code> | Für Redis-Read Replicas: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden bei <code>client-output-buffer-limit-replica-soft-limit</code> Bytes verbleibt, wird die Verbindung zum Client getrennt. |

| Name | Details | Beschreibung |
|------------------------|---|---|
| replica-allow-chaining | <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 über eigene Read Replicas verfügen darf.</p> |
| min-replicas-to-write | <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 Replikat unter diesen Wert abfällt, dann akzeptiert der primäre Knoten keine Schreibanforderungen mehr.</p> <p>Wenn entweder dieser Parameter oder 0 min-replicas-max-lag ist, akzeptiert der primäre Knoten immer Schreibanforderungen, auch wenn keine Replikat 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 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> |

Redis 4.0.10-Parameteränderungen

Familie der Parametergruppe: redis4.0

Redis 4.0.x-Standardparametergruppen

- `default.redis4.0` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus deaktiviert) und Replikationsgruppen.
- `default.redis4.0.cluster.on` – Verwenden Sie diese oder eine davon abgeleitete Parametergruppe für Redis-Cluster (Cluster-Modus aktiviert) und Replikationsgruppen.

In Redis 4.0.10 geänderte Parameter

| Name | Details | Beschreibung |
|-------------------------------|---|--|
| <code>maxmemory-policy</code> | <p>Gültige Werte: <code>allkeys-lru</code> , <code>volatile-lru</code> , <code>allkeys-lfu</code> , <code>volatile-lfu</code> , <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> , womit jeder beliebige Schlüssel mittels angenähertem LFU-Wert bereinigt wird, und <code>volatile-lfu</code> , womit unter den Schlüssel n mit einem festgelegten Ablaufdatum mittels angenährtem LFU-Wert bereinigt wird. In Version 6.2, als die R6gd-Knotenfamilie zur Verwendung mit Daten-Tiering eingeführt wurde, werden nur die maxmemory-Richtlinien <code>noeviction</code> , <code>volatile-lru</code> und <code>allkeys-lru</code> mit R6gd-Knotentypen unterstützt.</p> |

In Redis 4.0.10 hinzugefügte Parameter

| Name | Details | Beschreibung |
|-----------------------------------|---------|--------------|
| Parameter für asynchrones Löschen | | |

| Name | Details | Beschreibung |
|---------------------------------------|---|---|
| <code>lazyfree-lazy- eviction</code> | Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort | Führt eine asynchrone Löschung bei Bereinigungen durch. |
| <code>lazyfree-lazy-expire</code> | Zulässige Werte: yes/no Standard: no Typ: Boolesch Veränderbar: Ja Änderungen treten in Kraft: sofort | Führt eine asynchrone Löschung bei abgelaufenen Schlüsseln durch. |
| <code>lazyfree-lazy-server-del</code> | 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> |
| LFU-Parameter | | |
| <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 als CPU-Prozentsatz. |
| <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 als CPU-Prozentsatz. |

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. |

Redis 3.2.10-Parameteränderungen

Familie der Parametergruppe: redis3.2

ElastiCache für Redis 3.2.10 werden keine zusätzlichen Parameter unterstützt.

Redis 3.2.6-Parameteränderungen

Familie der Parametergruppe: redis3.2

Bei Redis 3.2.6 werden keine zusätzlichen Parameter unterstützt.

Redis 3.2.4-Parameteränderungen

Familie der Parametergruppe: redis3.2

Ab Redis 3.2.4 sind zwei Standard-Parametergruppen vorhanden.

- `default.redis3.2` – Wenn Sie Redis 3.2.4 einsetzen, geben Sie diese Parametergruppe oder eine davon abgeleitete Gruppe an, wenn Sie eine Redis-Replikationsgruppe (Cluster-Modus deaktiviert) erstellen und trotzdem die zusätzlichen Funktionen von Redis 3.2.4 nutzen möchten.
- `default.redis3.2.cluster.on` – Geben Sie diese Parametergruppe oder eine davon abgeleitete an, wenn Sie eine Redis-Replikationsgruppe (Cluster-Modus aktiviert) erstellen möchten.

Themen

- [Neue Parameters für Redis 3.2.4](#)
- [In Redis 3.2.4 \(Enhanced\) geänderte Parameter](#)

Neue Parameters für Redis 3.2.4

Familie der Parametergruppe: `redis3.2`

Bei Redis 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"> • -5: max. Größe: 64 Kb – wird nicht für normale Arbeitslasten empfohlen • -4: max. Größe: 32 Kb – wird nicht empfohlen • -3: max. Größe: 16 Kb – wird nicht empfohlen |

| Name | Details | Beschreibung |
|------|---------|---|
| | | <ul style="list-style-type: none">• -2: max. Größe: 8 Kb – wird empfohlen• -1: max. Größe: 4 Kb – wird empfohlen• Positive Zahlen bedeuten, bis hin zu genau dieser Anzahl von Elementen pro Listenknoten zu speichern. |

| 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"> • 0: Gesamte Komprimierung deaktivieren. • 1: Komprimierung jeweils einen Knoten nach innen vom Anfang und Ende starten. <pre>[head]->node->node->...->node->[tail]</pre> Alle Knoten außer [head] und [tail] werden komprimiert. • 2: Komprimierung jeweils zwei Knoten nach innen vom Anfang und Ende starten. <pre>[head]->[next]->node->node->...->node->[prev]->[tail]</pre> [head], [next], [prev], [tail] werden nicht komprimiert. Alle anderen Knoten werden komprimiert. • usw. |

| Name | Details | Beschreibung |
|------------------------------|--|---|
| <code>cluster-enabled</code> | Standard: no/yes * Typ: Zeichenfolge Veränderbar: Nein | <p>Gibt an, ob es sich um eine (Clustermodus-aktivierte) Redis-Replikationsgruppe im Cluster-Modus (ja) oder eine Redis-Replikationsgruppe (Cluster-Modus aktiviert) im Nicht-Cluster-Modus (Clustermodus) handelt. Redis (Cluster-Modus aktiviert) Replikationsgruppen im Cluster-Modus können ihre Daten über bis zu 500 Knotengruppen partitionieren.</p> <p>* Redis 3.2.x verfügt über zwei Standard-Parametergruppen.</p> <ul style="list-style-type: none">• <code>default.redis3.2</code> – Standardwert no.• <code>default.redis3.2.cluster.on</code> – Standardwert yes. <p>.</p> |

| Name | Details | Beschreibung |
|--|---|---|
| <code>cluster-require-full-coverage</code> | Standard: no Typ: Boolesch Veränderbar: Ja Änderungen werden wirksam: Sofort | <p>Wenn dieser Wert auf <code>yes</code> gesetzt ist, nehmen Redis-Knoten im Cluster-Modus (Cluster-Modus aktiviert) keine Anfragen mehr an, wenn sie feststellen, dass mindestens ein Hash-Slot unbesetzt 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>Es wird ein Wert von ca. 3 000 empfohlen, um die Vorteile der platzsparenden Codierung zu nutzen, ohne PFADD zu sehr zu verlangsamen. Dies entspricht der Einstellung O(N) bei der Sparse-Codierung. Der Wert kann auf ~10000 angehoben werden, wenn die CPU kein Problem darstellt, sondern der Speicherplatz, und der Datensatz besteht aus vielen Daten HyperLogLogs mit einer Kardinalität zwischen 0 und 15000.</p> |

| Name | Details | Beschreibung |
|--------------------------------------|--|--|
| <code>reserved-memory-percent</code> | Standard: 25 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort | <p>Der reservierte Prozentsatz des Knotenspeichers, der nicht für Daten verwendet werden soll. Standardmäßig wächst der Redis-Datenumfang, bis der gesamte Speicher des Knotens aufgebraucht ist. In diesem Fall wird die Knotenleistung aufgrund der übermäßigen Speicherauslagerung beeinträchtigt. Ein Teil des verfügbaren Speichers kann für andere Zwecke als für Redis reserviert werden, um den Auslagerungsaufwand zu verringern.</p> <p>Dieser Parameter ist spezifisch für die ElastiCache Redis-Standarddistribution und nicht Teil dieser Distribution.</p> <p>Weitere Informationen finden Sie unter <code>reserved-memory</code> und Verwalten von reserviertem Speicher.</p> |

In Redis 3.2.4 (Enhanced) geänderte Parameter

Familie der Parametergruppe: redis3.2

Bei Redis 3.2.4 wurden die folgenden Parameter geändert.

| Name | Details | Änderung |
|-------------------------------|--|-----------------------|
| <code>activeresharding</code> | Veränderbar: Ja, wenn die Parametergruppe keinem Cache-Cluster zugeordnet ist. Ansonsten nein. | Veränderbar war Nein. |
| <code>databases</code> | Veränderbar: Ja, wenn die Parametergruppe keinem | Veränderbar war Nein. |

| Name | Details | Änderung |
|-------------------------|--|---|
| | Cache-Cluster zugeordnet ist. Ansonsten nein. | |
| appendonly | Standard: deaktiviert Veränderbar: Nein | Wenn Sie von einer früheren Redis-Version aktualisieren möchten, müssen Sie zuerst <code>appendonly</code> deaktivieren. |
| appendfsync | Standard: deaktiviert Veränderbar: Nein | Wenn Sie von einer früheren Redis-Version aktualisieren möchten, müssen Sie zuerst <code>appendfsync</code> deaktivieren. |
| repl-timeout | Standard: 60 Veränderbar: Nein | Ist nun nicht mehr veränderbar mit einem Standard von 60. |
| tcp-keepalive | Standard: 300 | Standard war 0. |
| list-maxziplist-entries | | Der Parameter ist nicht mehr verfügbar. |
| list-maxziplist-value | | Der Parameter ist nicht mehr verfügbar. |

In Redis 2.8.24 (Enhanced) hinzugefügte Parameter

Familie der Parametergruppe: redis2.8

Bei Redis 2.8.24 werden keine zusätzlichen Parameter unterstützt.

In Redis 2.8.23 (Enhanced) hinzugefügte Parameter

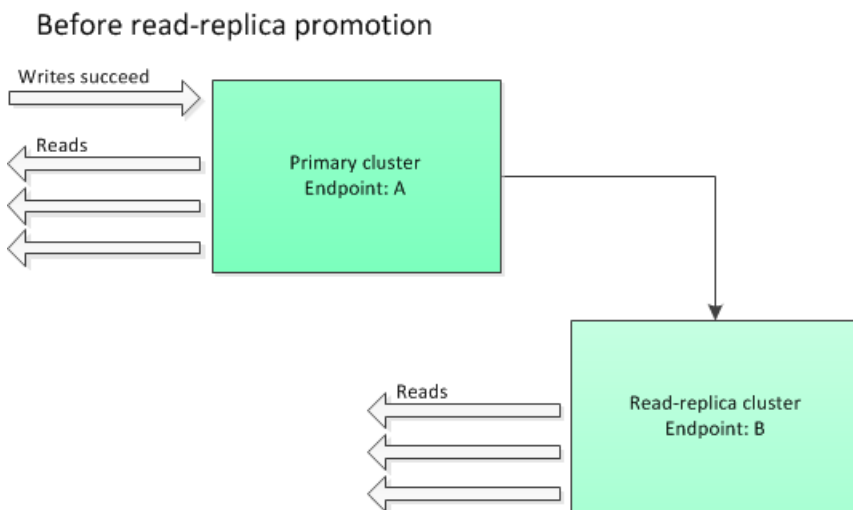
Familie der Parametergruppe: redis2.8

Bei Redis 2.8.23 wird der folgende zusätzliche Parameter unterstützt.

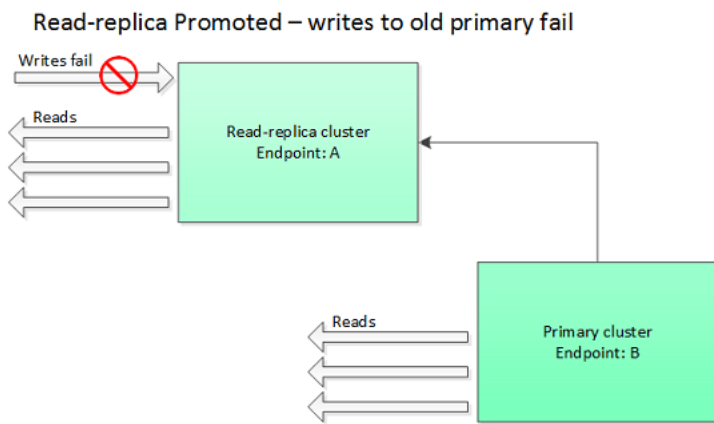
| Name | Details | Beschreibung |
|-----------------------------------|--|--|
| <code>close-on-slave-write</code> | Standard: <code>yes</code> Typ: Zeichenfolge (<code>yes/no</code>) Veränderbar: Ja Änderungen werden wirksam: Sofort | Wenn er aktiviert ist, wird die Verbindung mit Clients, die versuchen, in ein geschütztes Replikat zu schreiben, unterbrochen. |

Wie funktioniert `close-on-slave-write`

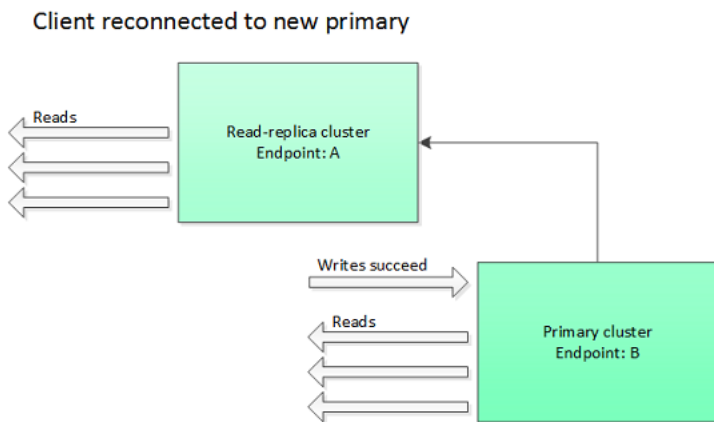
Der `close-on-slave-write` Parameter wurde von Amazon eingeführt, ElastiCache um Ihnen mehr Kontrolle darüber zu geben, wie Ihr Cluster reagiert, wenn ein primärer Knoten und ein Read Replica-Knoten aufgrund der Heraufstufung einer Read Replica zum primären Knoten die Rollen wechseln.



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 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. Die Logik Ihrer Anwendung sollte die Unterbrechung erkennen, die DNS-Tabelle überprüfen und wieder eine Verbindung zum primären Endpunkt herstellen, wobei es sich um Endpunkt B handeln würde.



Wann könnten Sie deaktivieren `close-on-replica-write`

Wenn `close-on-replica-write-Results` 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.

In Redis 2.8.22 (Enhanced) hinzugefügte Parameter

Familie der Parametergruppe: redis2.8

Bei Redis 2.8.22 werden keine zusätzlichen Parameter unterstützt.

Important

- Ab Redis-Version 2.8.22 gilt `repl-backlog-size` sowohl für den primären Cluster als auch für Replikat-Cluster.
- Ab Redis-Version 2.8.22 wird der Parameter `repl-timeout` nicht unterstützt. Wenn es geändert wird, ElastiCache wird es mit der Standardeinstellung (60s) überschrieben, wie wir es bei `tun.appendonly`

Die folgenden Parameter werden nicht mehr unterstützt.

- `appendonly`
- `appendfsync`
- `repl-timeout`

In Redis 2.8.21 hinzugefügte Parameter

Familie der Parametergruppe: redis2.8

Für Redis 2.8.21 werden keine zusätzlichen Parameter unterstützt.

In Redis 2.8.19 hinzugefügte Parameter

Familie der Parametergruppe: redis2.8

Bei Redis 2.8.19 werden keine zusätzlichen Parameter unterstützt.

In Redis 2.8.6 hinzugefügte Parameter

Familie der Parametergruppe: redis2.8


Bei Redis 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 Keyspace-Ereignisse, über die Redis Clients benachrichtigen kann. Jede Art von Ereignis wird durch einen einzelnen Buchstaben repräsentiert:</p> <ul style="list-style-type: none"> • K— Keyspace-Ereignisse, die mit dem Präfix __keyspace@<db> __ • E – Key-event-Ereignisse, die mit dem Präfix __keyevent@<db>__ veröffentlicht werden • g – Generische, unspezifische Befehle, wie z. B. DEL, EXPIRE, RENAME usw. • \$ – Zeichenfolgenbefehle • l — List-Befehle • s — Set-Befehle • h — Hash-Befehle • z — Sortierte Set-Befehle • x — Abgelaufene Ereignisse (Ereignisse, die jedes Mal generiert werden, wenn ein Schlüssel abläuft) • e — Ausgeschiedene Ereignisse (Ereignisse, die generiert |

| Name | Details | Beschreibung |
|------|---------|--|
| | | <p>werden, wenn ein Schlüssel für maxmemory entfernt wird)</p> <ul style="list-style-type: none">• A — Ein Alias für g\$!shzxe <p>Es kann eine beliebige Kombination dieser Ereignistypen vorliegen. So bedeutet AKE z. B., dass Redis Benachrichtigungen über alle Ereignistypen veröffentlichen 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 deaktiviert ist.</p> |

| Name | Details | Beschreibung |
|-------------------|---|---|
| repl-backlog-size | <p>Standard: 1048576</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p> | <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 1150 1507 1507" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Ab Redis-Version 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"> • Massendatenübertragung während der Synchronisierung, aus der Perspektive des Read Replica • Zeitüberschreitung des primären Knotens aus der Perspektive des Replikats • Zeitüberschreitung des Replikats aus der Perspektive des primären Knotens |

Parameter von Redis 2.6.13

Familie der Parametergruppe: redis2.6

Redis 2.6.13 war die erste Version von Redis, die von unterstützt wurde. ElastiCache Die folgende Tabelle zeigt die Redis 2.6.13-Parameter, die unterstützt werden. ElastiCache

| Name | Details | Beschreibung |
|---|---|--|
| <code>activeresha</code> <code>ashing</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. Der Hashvorgang für die Haupt-Hash-Tabelle wird zehnmal pro Sekunde erneut durchgeführt. Jeder erneut durchgeführte Hashvorgang verbraucht 1 Millisekunde der CPU-Zeit. Es empfiehlt sich, diesen Wert gering zu halten. Wenn einem Cluster eine neue Parameter |

| Name | Details | Beschreibung |
|------------|--|---|
| appendonly | Standard: no Typ: Zeichenfolge Veränderbar: Ja Änderungen werden wirksam: Sofort | <p>gruppe zugewiesen wird, muss dieser Wert in der alten und in der neuen Parametergruppe identisch sein.</p> <p>Aktiviert oder deaktiviert die Append-Only-File-Funktion (AOF) von Redis. AOF erfasst alle Redis-Befehle, die Daten im Cache ändern. Die Funktion wird zur Wiederherstellung bei bestimmten Knotenausfällen verwendet.</p> <p>Der Standardwert no bedeutet, dass AOF ausgeschaltet ist. Legen Sie für diesen Parameter yes fest, um AOF zu aktivieren.</p> <p>Weitere Informationen finden Sie unter Minimieren von Ausfällen.</p> <div data-bbox="829 957 1507 1318"><p> Note</p><p>Append-Only-Files (AOF) wird für cache.t1.micro- und cache.t2.*-Knoten nicht unterstützt. Für Knoten dieser Art wird der Parameterwert appendonly ignoriert.</p></div> <div data-bbox="829 1419 1507 1642"><p> Note</p><p>Für Multi-AZ-Replikationsgruppen ist AOF nicht zulässig.</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>Steuert, wenn <code>appendonly</code> auf „Yes“ (Ja) festgelegt, wie oft der AOF-Ausgabepuffer auf den Datenträger geschrieben wird:</p> <ul style="list-style-type: none"> • <code>no</code> – der Puffer wird nach Bedarf auf die Festplatte geschrieben. • <code>everysec</code> – Der Puffer wird einmal pro Sekunde geleert. Dies ist die Standardinstellung. • <code>always</code> (immer) – der Puffer wird jedes Mal geleert, wenn Daten im Cluster geändert werden. • <code>appendfsync</code> wird ab der Version 2.8.22 nicht mehr unterstützt. |
| <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 Redis-publish/subscribe-Clients: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Bytes 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 Redis-publish/subscribe-Clients: 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-pubsub-soft-seconds</code> lang andauert. |
| <code>client-output-buffer-limit-pubsub-soft-seconds</code> | Standard: 60 Typ: Ganzzahl Veränderbar: Ja Änderungen werden wirksam: Sofort | Für Redis-publish/subscribe-Clients: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden bei <code>client-output-buffer-limit-pubsub-soft-limit</code> Bytes verbleibt, wird die Verbindung zum Client getrennt. |

| Name | Details | Beschreibung |
|--|--|--|
| <code>client-output-buffer-limit-slave-hard-limit</code> | <p>Standard: Für Werte siehe Spezifische Parameter für den Redis-Knotentyp</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> | Für Redis-Read Replicas: Wenn der Ausgabepuffer eines Clients die angegebene Anzahl von Bytes erreicht, wird die Verbindung zum Client getrennt. |
| <code>client-output-buffer-limit-slave-soft-limit</code> | <p>Standard: Für Werte siehe Spezifische Parameter für den Redis-Knotentyp</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> | Für Redis-Read Replicas: 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-slave-soft-seconds</code> lang andauert. |
| <code>client-output-buffer-limit-slave-soft-seconds</code> | <p>Standard: 60</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Nein</p> | Für Redis-Read Replicas: Wenn der Ausgabepuffer eines Clients länger als diese Anzahl von Sekunden bei <code>client-output-buffer-limit-slave-soft-limit</code> Bytes 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 zum Stoppen des Skripts ergriffen werden.</p> <p>Wenn <code>lua-time-limit</code> überschritten wird, geben alle Redis-Befehle einen Fehler im Format <code>____-BUSY</code> zurück. Da dieser Status viele wichtige Redis-Operationen stören kann, ElastiCache wird zunächst ein <code>SCRIPT KILL</code>-Befehl ausgegeben. Wenn dies nicht erfolgreich ist, ElastiCache wird Redis zwangsweise neu gestartet.</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 Verwenden von Redis als LRU-Cache.</p> |

| Name | Details | Beschreibung |
|--------------------------------|---|--|
| <code>maxmemory-samples</code> | <p>Standard: 3</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p> | <p>Bei Berechnungen least-recently-used (LRU) und time-to-live (TTL) stellt dieser Parameter die Stichprobengröße der zu überprüfenden Schlüssel dar. Redis wählt standardmäßig 3 Schlüssel aus und verwendet den Schlüssel, der zuletzt verwendet wurde.</p> |
| <code>reserved-memory</code> | <p>Standard: 0</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p> | <p>Der Gesamtspeicher, in Byte, der nicht für Daten verwendet werden soll. Standardmäßig wächst der Redis-Knoten, bis der <code>maxmemory</code> des Knotens aufgebraucht ist (siehe Spezifische Parameter für den Redis-Knotentyp). In diesem Fall wird die Knotenleistung aufgrund der übermäßigen Speicherauslagerung beeinträchtigt. Ein Teil des verfügbaren Speichers kann für andere Zwecke als für Redis reserviert werden, um den Auslagerungsaufwand zu verringern.</p> <p>Dieser Parameter ist spezifisch für die ElastiCache Redis-Standarddistribution und nicht Teil der Standarddistribution.</p> <p>Weitere Informationen finden Sie unter <code>reserved-memory-percent</code> und Verwalten von reserviertem Speicher.</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 über eigene Read Replicas verfügen darf. |
| <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 Redis-Funktion „protokolliert“ werden sollen. |
| <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 „Slow Log“ von Redis. |

| 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> Important</p> <p>Einige Aspekte dieses Parameters haben sich in Redis-Version 3.2.4 geändert. Siehe In Redis 3.2.4 (Enhanced) geänderte Parameter.</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"> • 0 – Die Verbindung eines ungenutzten Client niemals trennen. • 1-19 – Ungültige Werte. • >=20 – Die Anzahl von Sekunden, die ein Knoten wartet, bevor die Verbindung eines ungenutzten Clients getrennt wird. |
| zset-maxziplist-entries | <p>Standard: 128</p> <p>Typ: Ganzzahl</p> <p>Veränderbar: Ja</p> <p>Änderungen werden wirksam: Sofort</p> | <p>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.</p> |

| Name | Details | Beschreibung |
|-------------------------------------|--|--|
| <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 für Ihren Redis 2.6.13-Cluster keine Parametergruppe angeben, wird die Standard-Parametergruppe verwendet (`default.redis2.6`). 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.

Spezifische Parameter für den Redis-Knotentyp

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-buff er-limit-slave-hard- limit | Client-output-buff er-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 |
| cache.m2.4xlarge | 70883737600 | 7088373760 | 7088373760 |
| cache.m3.medium | 2988441600 | 309329920 | 309329920 |
| cache.m3.large | 6501171200 | 650117120 | 650117120 |

| Knotentyp | Maxmemory | Client-output-buffer-limit-slave-hard-limit | Client-output-buffer-limit-slave-soft-limit |
|--------------------|--------------|---|---|
| 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 |
| cache.m6g.4xlarge | 56116178125 | 5611617812 | 5611617812 |
| cache.m6g.8xlarge | 111325552312 | 11132555231 | 11132555231 |
| cache.m6g.12xlarge | 168715971994 | 16871597199 | 16871597199 |

| Knotentyp | Maxmemory | Client-output-buffer-limit-slave-hard-limit | Client-output-buffer-limit-slave-soft-limit |
|--------------------|--------------|---|---|
| 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 |
| cache.r5.4xlarge | 113609865216 | 11360986522 | 11360986522 |
| cache.r5.12xlarge | 341206346547 | 34120634655 | 34120634655 |
| cache.r5.24xlarge | 682485973811 | 68248597381 | 68248597381 |

| Knotentyp | Maxmemory | Client-output-buffer-limit-slave-hard-limit | Client-output-buffer-limit-slave-soft-limit |
|---------------------|--------------|---|---|
| cache.r6g.large | 14037181030 | 1403718103 | 1403718103 |
| cache.r6g.xlarge | 28261849702 | 2826184970 | 2826184970 |
| dache.r6g.2xlarge | 56711183565 | 5671118356 | 5671118356 |
| dache.r6g.4xlarge | 113609865216 | 11360986522 | 11360986522 |
| dache.r6g.8xlarge | 225000375228 | 22500037523 | 22500037523 |
| dache.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 |
| cache.r7g.4xlarge | 113609865216 | 11360986522 | 11360986522 |
| cache.r7g.8xlarge | 225000375228 | 22500037523 | 22500037523 |
| cache.r7g.12xlarge | 341206346547 | 34120634655 | 34120634655 |

| Knotentyp | Maxmemory | Client-output-buff er-limit-slave-hard- limit | Client-output-buff er-limit-slave-soft- limit |
|---------------------|--------------|---|---|
| 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 Amazon Virtual Private Cloud VPC erstellt.

T1-Instances unterstützen Multi-AZ nicht.

T1- und T2-Instances unterstützen Redis AOF nicht.

Die Redis-Konfigurationsvariablen `appendonly` und `appendfsync` werden auf der Redis-Version 2.8.22 und höher nicht unterstützt.

Skalierung ElastiCache für Redis

Serverlose Skalierung ElastiCache

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 CPU, Arbeitsspeicher 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 für Ihre Anwendung kommt. Sie können die Ressourcen überwachen, die von Ihrem Cache-In CloudWatch verbraucht werden, indem Sie die `BytesUsedForCache` Metrik für den Cache-Datenspeicher und `ElastiCacheProcessingUnits` (ECPU) für die Computernutzung überwachen.

Festlegen von Skalierungslimits zur Kostenkontrolle

Sie können wählen, ob Sie eine maximale Nutzung sowohl für den Cache-Datenspeicher als auch für die 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 verringerten Cache-Leistung kommen, sobald der Cache das Maximum erreicht. Wenn Sie ein Maximum für den Cache-Datenspeicher festlegen und Ihr Cache-Datenspeicher das Maximum erreicht, ElastiCache werden mithilfe der LRU-Logik Daten aus Ihrem Cache gelöscht, für die eine Gültigkeitsdauer (TTL) festgelegt ist. Wenn es keine Daten gibt, die bereinigt werden können, wird bei Anfragen zum Schreiben zusätzlicher Daten eine Out-of-Memory-Fehlermeldung (OOM) angezeigt. Wenn Sie ein Maximum in ECPU/Sekunde festlegen und die Rechenauslastung Ihres Workloads diesen Wert überschreitet, beginnt die Drosselung von Redis-Anfragen. ElastiCache

Wenn Sie ein maximales Limit 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 mithilfe der ElastiCache Konsole, der CLI oder der API durchführen. ElastiCache Serverless aktualisiert die verfügbaren eCPUs/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/Sekunde oder Datenspeicher über die Konsole, CLI oder API aktualisiert wird, ist dieses neue Limit innerhalb von 1 Stunde verfügbar. ElastiCache Serverless unterstützt 30.000 ECPUs/Sekunde bei leerem Cache und bis zu 90.000 ECPUs/s bei Verwendung der Funktion „Aus Replikat lesen“. ElastiCache kann ECPUS/Sekunde alle 10–12 Minuten verdoppeln. 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 Spitzenwert pro Sekunde festzulegen, den Sie mindestens 60 Minuten vor dem 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 der neuen Mindestanzahl an 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 Minimum 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. Für die Nutzung von ECPU oder Datenspeicher, die das Mindestnutzungslimit überschreiten, 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 der ECPU-Preise 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 bei Verwendung von Read from Replica mit READONLY-Verbindungen). Als bewährte Methode sollte Ihre Anwendung sicherstellen, dass die Schlüsselverteilung auf die Redis-Steckplätze 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

Verwenden Sie die `modify-serverless-cache` API, um die Skalierungsgrenzen mithilfe der CLI zu ändern.

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 Skalierungslimits mithilfe der CLI

Um Skalierungsgrenzen mithilfe der CLI zu entfernen, setzen Sie die Parameter `Minimal` 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 ElastiCache für selbst entwickelte Redis-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. Durch die Verwendung von Amazon können ElastiCache Sie skalieren, um

den aktuellen Bedarf zu decken, und zahlen nur für das, was Sie nutzen. ElastiCache ermöglicht es Ihnen, Ihren Cache entsprechend dem Bedarf zu skalieren.

Im Folgenden finden Sie das richtige Thema für die Skalierungsaktionen, die Sie ausführen möchten.

Skalieren von Redis-Clustern

| Aktion | Redis-Cluster (Clustermodus deaktiviert) | Redis (Cluster-Modus aktiviert) |
|------------------------|---|--|
| Einskalieren | Entfernen von Knoten aus einem Cluster | Skalieren von Clustern in Redis (Clustermodus aktiviert) |
| Ausskalieren | Hinzufügen von Knoten zu einem Cluster | Online-Resharding und Shard-Rebalancing für Redis (Clustermodus aktiviert) |
| Ändern von Knotentypen | <p>Zu einem größeren Knotentyp:</p> <ul style="list-style-type: none"> Hochskalieren von Clustern mit einem Knoten für Redis (Clustermodus deaktiviert) Aufwärtsskalieren von Redis-Clustern mit Replikate n <p>Zu einem kleineren Knotentyp:</p> <ul style="list-style-type: none"> Abwärtsskalieren eines Redis-Clusters mit einem Knoten Abwärtsskalieren von Redis-Clustern mit Replikaten | Vertikales Online-Skalieren durch Ändern des Knotentyps |

| Aktion | Redis-Cluster (Clustermodus deaktiviert) | Redis (Cluster-Modus aktiviert) |
|-------------------------------------|---|--|
| Ändern der Anzahl der Knotengruppen | Nicht unterstützt für Redis-Cluster (Cluster-Modus deaktiviert) | Skalieren von Clustern in Redis (Clustermodus aktiviert) |

Themen

- [Skalieren von Clustern für Redis \(Clustermodus deaktiviert\)](#)
- [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#)

Skalieren von Clustern für Redis (Clustermodus deaktiviert)

Redis-Cluster (Cluster-Modus deaktiviert) können ein Single-Node-Cluster mit 0 Shards oder Multi-Node-Cluster mit 1 Shard sein. 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.

Inhalt

- [Skalieren von Clustern mit einem Knoten für Redis \(Clustermodus deaktiviert\)](#)
 - [Hochskalieren von Clustern mit einem Knoten für Redis \(Clustermodus deaktiviert\)](#)
 - [Hochskalieren von Clustern mit einem Knoten für Redis \(Clustermodus deaktiviert\) \(Konsole\)](#)
 - [Hochskalierung eines Redis-Cache-Clusters mit einem Knoten \(AWS CLI\)](#)
 - [Skalierung von Redis-Cache-Clustern mit einem Knoten \(ElastiCache-API\)](#)
 - [Abwärtsskalieren eines Redis-Clusters mit einem Knoten](#)
 - [Herunterskalierung eines Redis-Clusters mit einem Knoten \(Konsole\)](#)
 - [Herunterskalierung eines Redis-Cache-Clusters mit einem Knoten \(AWS CLI\)](#)
 - [Herunterskalierung eines Redis-Cache-Clusters mit einem Knoten \(ElastiCache-API\)](#)
- [Skalieren von Redis-Clustern \(Clustermodus deaktiviert\) mit Replikatknoten](#)
 - [Aufwärtsskalieren von Redis-Clustern mit Replikaten](#)
 - [Abwärtsskalieren von Redis-Clustern mit Replikaten](#)
 - [Erhöhen der Lesekapazität](#)
 - [Verringern der Lesekapazität](#)

Skalieren von Clustern mit einem Knoten für Redis (Clustermodus deaktiviert)

Die Redis-Knoten (Cluster-Modus deaktiviert) müssen groß genug sein, um alle Daten des Cache plus Redis-Overhead aufzunehmen. Um die Datenkapazität Ihres Redis-Clusters (Cluster-Modus deaktiviert) zu ändern, müssen Sie vertikal skalieren, d. h. auf einen größeren Knotentyp hochskalieren, um die Datenkapazität zu erhöhen, oder auf einen kleineren Knotentyp herunterskalieren, um die Datenkapazität zu verringern.

Der Skalierungsprozess von ElastiCache for Redis ist so konzipiert, dass die vorhandenen Daten so gut wie möglich erhalten bleiben und setzt eine erfolgreiche Redis-Replikation voraus. Bei Redis-

Clustern (Cluster-Modus deaktiviert) wird empfohlen, Redis ausreichend Speicher zur Verfügung zu stellen.

Sie können Ihre Daten nicht über mehrere Redis-Cluster (Cluster-Modus deaktiviert) verteilen. Wenn Sie jedoch nur die Lesekapazität Ihres Clusters erhöhen oder verringern müssen, können Sie einen Redis-Cluster (Cluster-Modus deaktiviert) mit Replikationsknoten erstellen und Lesereplikate hinzufügen oder entfernen. Um einen Redis-Cluster (Cluster-Modus deaktiviert) mit Replikatknoten zu erstellen, der Ihren Redis-Cache-Cluster mit einem Knoten als primären Cluster verwendet, siehe [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

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](#).

Neben der Möglichkeit, die Lesekapazität zu skalieren, bieten Redis-Cluster (Cluster-Modus deaktiviert) mit Replikaten weitere geschäftliche Vorteile. Weitere Informationen finden Sie unter [Hohe Verfügbarkeit mit Replikationsgruppen](#).

Important

Wenn Sie in Ihrer Parametergruppe mit `reserved-memory` Speicher für den Redis-Overhead reservieren, stellen Sie vor der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die korrekte Speichermenge für den neuen Knotentyp reserviert. 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 [Verwalten von reserviertem Speicher](#).

Themen

- [Hochskalieren von Clustern mit einem Knoten für Redis \(Clustermodus deaktiviert\)](#)
- [Abwärtsskalieren eines Redis-Clusters mit einem Knoten](#)

Hochskalieren von Clustern mit einem Knoten für Redis (Clustermodus deaktiviert)

Wenn Sie einen Redis-Cluster mit einem Knoten skalieren, führt ElastiCache folgende Aktionen unabhängig davon aus, ob Sie die ElastiCache-Konsole, das AWS CLI oder die ElastiCache-API verwenden.

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 Prozess 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 Schreiboperationen des Primärknotens 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, wird die Aufwärtsskalierung des Redis-Clusters blockiert, wenn für das nächste Wartungsfenster ein Engine-Upgrade geplant ist. Weitere Informationen zu Wartungsfenstern finden Sie unter [Verwaltung der Wartung](#).

Blockierte Redis-Operationen

| 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 Ihre Redis-Skalierung für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Apply immediately deaktivieren (CLI: `--no-apply-immediately`, API: `ApplyImmediately=false`).
- Warten Sie bis zum nächsten Wartungsfenster (oder länger), um Ihre Redis-Skalierung auszuführen.
- Fügen Sie dieser Cache-Cluster-Änderung das Redis-Engine-Upgrade durch Aktivieren des Kontrollkästchens Apply Immediately hinzu (CLI: `--apply-immediately`, API: `ApplyImmediately=true`). Dadurch wird die Skalierung freigegeben, da das Engine-Upgrade sofort ausgeführt wird.

Sie können einen Redis-Cluster (Cluster-Modus deaktiviert) mithilfe der ElastiCache Konsole, der AWS CLI oder der ElastiCache API.

Important

Wenn Sie in Ihrer Parametergruppe mit `reserved-memory` Speicher für den Redis-Overhead reservieren, stellen Sie vor der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die korrekte Speichermenge für den neuen Knotentyp reserviert. 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 [Verwalten von reserviertem Speicher](#).

Hochskalieren von Clustern mit einem Knoten für Redis (Clustermodus deaktiviert) (Konsole)

Im folgenden Verfahren wird beschrieben, wie Sie einen Redis-Cluster mit einem Knoten über die ElastiCache-Managementkonsole skalieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

So skalieren Sie einen Redis-Cluster mit einem Knoten (Konsole)

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis clusters (Redis-Cluster) aus.
3. Wählen Sie in der Liste der Cluster den gewünschten Cluster aus (er muss mit der Redis-Engine, nicht mit der Clustered Redis-Engine betrieben werden).
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.

Hochskalierung eines Redis-Cache-Clusters mit einem Knoten (AWS CLI)

Im folgenden Verfahren wird beschrieben, wie Sie einen Redis-Cache-Cluster mit einem Knoten über die AWS CLI skalieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

So skalieren Sie einen Redis-Cache-Cluster mit einem Knoten (AWS CLI)

1. Führen Sie den AWS CLI `list-allowed-node-type-modifications`-Befehl mit dem folgenden Parameter aus, um die Knotentypen zu bestimmen, die für eine Aufwärtsskalierung verfügbar sind.
 - `--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 in etwa folgendermaßen aus (JSON format).

```
{  
  "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-modifications](#) in der AWS CLI-Referenz.

2. Geben Sie mit dem AWS CLI `modify-cache-cluster`-Befehl und den folgenden Parametern den zu skalierenden Cache-Cluster und den neuen, größeren Knotentyp an, um den vorhandenen Cache-Cluster zu ändern.
- `--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 in etwa folgendermaßen aus (JSON format).

```
{  
  "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 die `--apply-immediately` verwenden, prüfen Sie mit dem AWS CLI `describe-cache-clusters`-Befehl mit dem folgenden Parameter den Status des neuen Cache-Clusters. Wenn der Status zu Verfügbar wechselt, können Sie den neuen, größeren Cache-Cluster verwenden.
- `--cache-cluster-id` `cluster-id` Der Name des 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 AWS CLI-Referenz.

Skalierung von Redis-Cache-Clustern mit einem Knoten (ElastiCache-API)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe einer ElastiCache-API einen Redis-Cache-Cluster mit einem Knoten aufskalieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

So skalieren Sie einen Redis-Cache-Cluster mit einem Knoten auf (ElastiCache-API)

1. Bestimmen Sie die Knotentypen, auf die Sie hochskalieren können, indem Sie die ElastiCache-API-Aktion `ListAllowedNodeTypeModifications` mit dem folgenden Parameter ausführen.
 - `CacheClusterId` Der Name des Redis-Cache-Clusters mit einem Knoten, den Sie skalieren 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 Amazon-ElastiCache-API-Referenz.

2. Ändern Sie Ihren vorhandenen Cache-Cluster unter Angabe des zu vergrößernden Cache-Clusters und des neuen, größeren Knotentyps, indem Sie die ElastiCache-API-Aktion `ModifyCacheCluster` und die folgenden Parameter verwenden.
 - `CacheClusterId` – Der Name des aufzuskalierenden Cache-Clusters.
 - `CacheNodeType` Der neue, größere Knotentyp, auf den der Cache-Cluster skaliert werden soll. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Aufruf `ListAllowedNodeTypeModifications` 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 Amazon-ElastiCache-API-Referenz.

3. Wenn Sie `ApplyImmediately=true` verwenden, prüfen Sie mit dem ElastiCache-API-Aufruf `DescribeCacheClusters` mit dem folgenden Parameter den Status des neuen Cache Clusters. Wenn der Status zu `Verfügbar` wechselt, können Sie den neuen, größeren Cache-Cluster verwenden.
- `CacheClusterId` Der Name des 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 Amazon-ElastiCache-API-Referenz.

Abwärtsskalieren eines Redis-Clusters mit einem Knoten

In den folgenden Abschnitten wird erläutert, wie Sie einen Redis-Cluster mit einem Knoten auf einen kleineren Knotentyp skalieren. Es ist für den langfristigen Erfolg Ihres neuen Redis-Clusters wichtig sicherzustellen, dass der neue, kleinere Knotentyp für die Datenmenge und den Redis-Overhead ausreicht. Weitere Informationen finden Sie unter [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen 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 Redis-Clusters mit einem Knoten \(Konsole\)](#)
- [Herunterskalierung eines Redis-Cache-Clusters mit einem Knoten \(AWS CLI\)](#)
- [Herunterskalierung eines Redis-Cache-Clusters mit einem Knoten \(ElastiCache-API\)](#)

Herunterskalierung eines Redis-Clusters mit einem Knoten (Konsole)

Im folgenden Verfahren wird erläutert, wie Sie einen Redis-Cluster mit einem Knoten mithilfe der ElastiCache-Konsole auf einen kleineren Knotentyp skalieren.

Important

Wenn Sie in Ihrer Parametergruppe mit `reserved-memory` Speicher für den Redis-Overhead reservieren, stellen Sie vor der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die korrekte Speichermenge für den neuen Knotentyp reserviert. 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 [Verwalten von reserviertem Speicher](#).

So skalieren Sie einen Redis-Cluster mit einem Knoten herunter (Konsole)

1. Vergewissern Sie sich, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht.
2. Wenn Sie in Ihrer Parametergruppe mit `reserved-memory` Speicher für den Redis-Overhead reservieren, stellen Sie sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die korrekte Speichermenge für den neuen Knotentyp reserviert.

Alternativ können Sie eine benutzerdefinierte Parametergruppe für die Verwendung von `reserved-memory-percent` anpassen. Weitere Informationen finden Sie unter [Verwalten von reserviertem Speicher](#).

3. Melden Sie sich bei AWS Management Console an 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. Der Cluster muss mit der Redis-Engine und nicht mit der Clustered Redis-Engine betrieben werden.
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 eines Redis-Cache-Clusters mit einem Knoten (AWS CLI)

Im folgenden Verfahren wird beschrieben, wie Sie einen Redis-Cache-Cluster mit einem Knoten über die AWS CLI skalieren.

So skalieren Sie einen Redis-Cache-Cluster mit einem Knoten herunter (AWS CLI)

1. Führen Sie den AWS CLI-Befehl `list-allowed-node-type-modifications` mit dem folgenden Parameter aus, um die Knotentypen zu bestimmen, die für eine Abwärtsskalierung verfügbar sind.

- `--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 in etwa folgendermaßen aus (JSON format).

```
{  
  "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-modifications](#) in der AWS CLI-Referenz.

2. Geben Sie mit dem AWS CLI-Befehl `modify-cache-cluster` und den folgenden Parametern den zu skalierenden Cache-Cluster und den neuen, kleineren Knotentyp an, um den vorhandenen Cache-Cluster zu ändern.
 - `--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 in etwa folgendermaßen aus (JSON format).

```
{
  "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 die `--apply-immediately` verwenden, prüfen Sie mit dem AWS CLI `describe-cache-clusters`-Befehl mit dem folgenden Parameter den Status des neuen Cache-Clusters.

Wenn der Status zu Verfügbar wechselt, können Sie den neuen, größeren Cache-Cluster verwenden.

- `--cache-cache cluster-id` Der Name des 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 AWS CLI-Referenz.

Herunterskalierung eines Redis-Cache-Clusters mit einem Knoten (ElastiCache-API)

Im folgenden Verfahren wird beschrieben, wie Sie mithilfe einer ElastiCache-API einen Redis-Cache-Cluster mit einem Knoten herunterskalieren.

So skalieren Sie einen Redis-Cache-Cluster mit einem Knoten ab (ElastiCache-API)

1. Bestimmen Sie die Knotentypen, auf die Sie herunterskalieren können, indem Sie die ElastiCache-API-Aktion `ListAllowedNodeTypeModifications` mit dem folgenden Parameter ausführen.
 - `CacheClusterId` – Der Name des 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 Amazon-ElastiCache-API-Referenz.

2. Ändern Sie Ihren vorhandenen Cache-Cluster unter Angabe des zu vergrößernden Cache-Clusters und des neuen, größeren Knotentyps, indem Sie die ElastiCache-API-Aktion `ModifyCacheCluster` und die folgenden Parameter verwenden.
 - `CacheClusterId` – Der Name des herunter zu skalierenden Cache-Clusters.
 - `CacheNodeType` – Der neue Knotentyp, auf den der Cache-Cluster herunterskaliert werden soll. Der Wert muss einer der Knotentypen sein, die in Schritt 1 mit dem Aufruf `ListAllowedNodeTypeModifications` 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-xl  
  &Version=2015-02-02  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150202T192317Z  
  &X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie unter [ModifyCacheCluster](#) in der Amazon-ElastiCache-API-Referenz.

3. Wenn Sie `ApplyImmediately=true` verwenden, prüfen Sie mit dem ElastiCache-API-Aufruf `DescribeCacheClusters` mit dem folgenden Parameter den Status des neuen Cache Clusters. Wenn der Status zu verfügbar wechselt, können Sie den neuen, kleineren Cache-Cluster verwenden.

- `CacheClusterId` Der Name des 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 Amazon-ElastiCache-API-Referenz.

Skalieren von Redis-Clustern (Clustermodus deaktiviert) mit Replikatknoten

Ein Redis-Cluster mit Replikatknoten (in der API/CLI als Replikationsgruppe bezeichnet) ist über Replikation mit Multi-AZ mit automatischem Failover hochverfügbar. Ein Cluster mit Replikatknoten ist eine logische Sammlung von bis zu sechs Redis-Knoten, wobei ein Knoten, der Primärknoten, sowohl Lese- als auch Schreibanfragen verarbeiten 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 Redis (Cluster-Modus deaktiviert) die Partitionierung Ihrer Daten über mehrere Cluster nicht unterstützt, enthält jeder Knoten in einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert) das gesamte Cache-Dataset. Redis (Cluster-Modus aktiviert) -Cluster unterstützen die Partitionierung Ihrer Daten über 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.

Die Aufwärtsskalierung in ElastiCache ist so ausgelegt, dass Ihre vorhandenen Daten möglichst beibehalten werden. Außerdem erfordert sie eine erfolgreiche Redis-Replikation. Für Redis-Cluster mit Replikaten sollte Redis ausreichend Speicher zur Verfügung gestellt werden.

Verwandte Themen

- [Hohe Verfügbarkeit mit Replikationsgruppen](#)
- [Replikation: Redis \(Cluster-Modus deaktiviert\) im Vergleich zu Redis \(Cluster-Modus aktiviert\)](#)
- [Minimierung der Ausfallzeiten ElastiCache bei Redis mit Multi-AZ](#)
- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#)

Themen

- [Aufwärtsskalieren von Redis-Clustern mit Replikaten](#)
- [Abwärtsskalieren von Redis-Clustern mit Replikaten](#)
- [Erhöhen der Lesekapazität](#)
- [Verringern der Lesekapazität](#)

Aufwärtsskalieren von Redis-Clustern mit Replikaten

Amazon ElastiCache bietet Unterstützung für die Konsole, CLI und API zur Skalierung Ihrer Redis-Replikationsgruppe (Cluster-Modus deaktiviert).

Beim Starten der Skalierung tut ElastiCache Folgendes:

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. Die DNS-Einträge werden aktualisiert und verweisen nun auf die neuen Knoten. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Bei Redis 5.0 und höher können Sie zum automatischen Failover fähige Cluster skalieren, während der Cluster online bleibt und eingehende Anforderungen ausführt. Bei Version 4.0.10 und niedriger stellen Sie bei vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird.
5. Die alten Knoten (CLI/API: Replikationsgruppe) werden gelöscht. 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, wird die Aufwärtsskalierung des Redis-Clusters blockiert, wenn für das nächste Wartungsfenster des Clusters ein Engine-Upgrade geplant ist.

Blockierte Redis-Operationen

| 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 Ihre Redis-Skalierung für das nächste Wartungsfenster, indem Sie das Kontrollkästchen Apply immediately deaktivieren (CLI: `--no-apply-immediately`, API: `ApplyImmediately=false`).
- Warten Sie bis zum nächsten Wartungsfenster (oder länger), um Ihre Redis-Skalierung auszuführen.
- Fügen Sie dieser Cache-Cluster-Änderung das Redis-Engine-Upgrade durch Aktivieren des Kontrollkästchens Apply Immediately hinzu (CLI: `--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 einen Redis-Cluster mit Replikaten mithilfe der ElastiCache-Konsole, dem AWS CLI und der ElastiCache-API hochskalieren.

Important

Wenn Sie in Ihrer Parametergruppe mit `reserved-memory` Speicher für den Redis-Overhead reservieren, stellen Sie vor der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die korrekte Speichermenge für den neuen Knotentyp reserviert. 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 [Verwalten von reserviertem Speicher](#).

Hochskalierung von Redis-Clustern mit Replikaten (Konsole)

Wie lange die Aufwärtsskalierung auf einen größeren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cluster ab.

Im nachfolgenden Verfahren wird ein Cluster mit Replikaten mithilfe der ElastiCache-Konsole vom derzeitigen Knotentyp auf einen neuen, größeren Knotentyp skaliert. Während dieses Vorgangs kann für andere Versionen eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten auftreten, während der DNS-Eintrag aktualisiert wird. Für Knoten, die auf 5.0.6-Versionen ausgeführt werden, tritt möglicherweise eine Ausfallzeit unter 1 Sekunde und bei älteren Versionen von wenigen Sekunden auf.

So skalieren Sie einen Redis-Cluster mit Replikaten hoch (Konsole)

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis clusters (Redis-Cluster) aus.
3. Wählen Sie aus der Liste der Cluster den Cluster aus, den Sie skalieren möchten. Der Cluster muss mit der Redis-Engine und nicht mit der Clustered Redis-Engine betrieben werden.
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.

Aufwärtsskalieren einer Redis-Replikationsgruppe (AWS CLI)

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der AWS CLI vom derzeitigen Knotentyp auf einen neuen, größeren Knotentyp skaliert. Während dieses Prozesses aktualisiert ElastiCache für Redis die DNS-Einträge, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Bei Redis 5.0 und höher können Sie zum automatischen Failover fähige Cluster skalieren, während der Cluster online bleibt und eingehende Anforderungen ausführt. Bei Version 4.0.10 und niedriger stellen Sie bei vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird.

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 Redis-Replikationsgruppe hoch (AWS CLI)

1. Führen Sie den AWS CLI `list-allowed-node-type-modifications`-Befehl mit dem folgenden Parameter aus, um die Knotentypen zu bestimmen, die für eine Aufwärtsskalierung verfügbar sind.
 - `--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 sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{  
  "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-modifications](#) in der AWS CLI-Referenz.

2. Führen Sie den AWS CLI `modify-replication-group`-Befehl mit dem folgenden Parameter aus, um die derzeitige Replikationsgruppe auf den neuen Knotentyp zu skalieren.
 - `--replication-group-id` der Name der Replikationsgruppe.
 - `--cache-node-type` der neue, größere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Der Wert muss einer der Instance-Typen 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 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 sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
```

```
"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 Parameter `--apply-immediately` verwenden, überwachen Sie mit dem AWS CLI `describe-replication-group`-Befehl mit dem folgenden Parameter den Status der Replikationsgruppe. Wenn der Status noch wird geändert lautet, tritt möglicherweise eine Ausfallzeit unter 1 Sekunde für Knoten auf, die auf 5.0.6-Versionen ausgeführt werden, und für ältere Versionen eine kurze Unterbrechung von Lese- und Schreibvorgängen vom Primärknoten, 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 unter [describe-replication-groups](#) in der AWS CLI-Referenz.

Skalieren einer Redis-Replikationsgruppe (ElastiCache API)

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der ElastiCache-API vom derzeitigen Knotentyp auf einen neuen, größeren Knotentyp skaliert. Bei Redis 5.0 und höher können Sie zum automatischen Failover fähige Cluster skalieren, während der Cluster online bleibt und eingehende Anforderungen ausführt. Bei Version 4.0.10 und niedriger stellen Sie bei vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird.

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 Redis-Replikationsgruppe hoch (ElastiCache-API)

1. Führen Sie die ElastiCache-API-Aktion `ListAllowedNodeTypeModifications` mit dem folgenden Parameter aus, um die Knotentypen zu bestimmen, die für eine Aufwärtsskalierung verfügbar sind.

- `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 Amazon-ElastiCache-API-Referenz.

2. Führen Sie die ElastiCache-API-Aktion `ModifyRedplicationGroup` mit dem folgenden Parameter aus, um die derzeitige Replikationsgruppe auf den neuen Knotentyp zu skalieren.
 - `ReplicationGroupId` der Name der Replikationsgruppe.
 - `CacheNodeType` der neue, größere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Der Wert muss einer der Instance-Typen sein, die in Schritt 1 mit dem Aufruf `ListAllowedNodeTypeModifications` 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 Amazon-ElastiCache-API-Referenz.

3. Wenn Sie `ApplyImmediately=true` verwenden, überwachen Sie mit der ElastiCache-API-Aktion `DescribeReplicationGroups` mit den folgenden Parametern den Status der Replikationsgruppe. 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 Amazon-ElastiCache-API-Referenz.

Abwärtsskalieren von Redis-Clustern mit Replikaten

In den folgenden Abschnitten wird erläutert, wie Sie einen Redis-Cache-Cluster (Cluster-Modus deaktiviert) mit Replikatknoten auf einen kleineren Knotentyp skalieren. 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 Arbeitsspeicher verfügen, um einen 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 Sie in Ihrer Parametergruppe mit `reserved-memory` Speicher für den Redis-Overhead reservieren, stellen Sie vor der Skalierung sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die korrekte Speichermenge für den neuen Knotentyp reserviert. 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 [Verwalten von reserviertem Speicher](#).

Herunterskalierung eines Redis-Clusters mit Replikaten (Konsole)

Im nachfolgenden Verfahren wird ein Redis-Cluster mit Replikatknoten mithilfe der ElastiCache-Konsole auf einen kleineren Knotentyp skaliert.

So skalieren Sie einen Redis-Cluster mit Replikatknoten herunter (Konsole)

1. Vergewissern Sie sich, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht.
2. Wenn Sie in Ihrer Parametergruppe mit `reserved-memory` Speicher für den Redis-Overhead reservieren, stellen Sie sicher, dass Sie über eine benutzerdefinierte Parametergruppe verfügen, die die korrekte Speichermenge für den neuen Knotentyp reserviert.

Alternativ können Sie eine benutzerdefinierte Parametergruppe für die Verwendung von `reserved-memory-percent` anpassen. Weitere Informationen finden Sie unter [Verwalten von reserviertem Speicher](#).

3. Melden Sie sich bei AWS Management Console an 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. Der Cluster muss mit der Redis-Engine und nicht mit der Clustered Redis-Engine betrieben werden.
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.

Abwärtsskalieren einer Redis-Replikationsgruppe (AWS CLI)

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der AWS CLI vom derzeitigen Knotentyp auf einen neuen, kleineren Knotentyp skaliert. Während dieses Prozesses aktualisiert ElastiCache für Redis die DNS-Einträge, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Bei Redis 5.0 und höher können Sie zum automatischen Failover fähige Cluster skalieren, während der Cluster online bleibt und eingehende Anforderungen ausführt. Bei Version 4.0.10 und niedriger stellen Sie bei vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird.

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.

Abwärtsskalieren einer Redis-Replikationsgruppe (AWS CLI)

1. Führen Sie den AWS CLI-Befehl `list-allowed-node-type-modifications` mit dem folgenden Parameter aus, um die Knotentypen zu bestimmen, die für eine Abwärtsskalierung verfügbar sind.
 - `--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 sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{  
  "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-modifications](#) in der AWS CLI-Referenz.

2. Führen Sie den AWS CLI `modify-replication-group`-Befehl mit dem folgenden Parameter aus, um die derzeitige Replikationsgruppe auf den neuen Knotentyp zu skalieren.

- `--replication-group-id` der Name der Replikationsgruppe.
- `--cache-node-type` Der neue, kleinere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Der Wert muss einer der Instance-Typen 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 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 sollte etwa wie nachfolgend dargestellt aussehen (JSON-Format).

```
{
  "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 Parameter `--apply-immediately` verwenden, überwachen Sie mit dem AWS CLI `describe-replication-group`-Befehl mit dem folgenden Parameter den Status der Replikationsgruppe. 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 unter [describe-replication-groups](#) in der AWS CLI-Referenz.

Herunterskalierung einer Redis-Replikationsgruppe (ElastiCache-API)

Im nachfolgenden Verfahren wird eine Replikationsgruppe mithilfe der ElastiCache-API vom derzeitigen Knotentyp auf einen neuen, kleineren Knotentyp skaliert. Während dieses Prozesses aktualisiert ElastiCache for Redis die DNS-Einträge, sodass sie auf die neuen Knoten verweisen.

Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Bei Redis 5.0 und höher können Sie zum automatischen Failover fähige Cluster skalieren, während der Cluster online bleibt und eingehende Anforderungen ausführt. Bei Version 4.0.10 und niedriger stellen Sie bei vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird. 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.

Herunterskalierung einer Redis-Replikationsgruppe (ElastiCache-API)

1. Führen Sie die ElastiCache-API-Aktion `ListAllowedNodeTypeModifications` mit dem folgenden Parameter aus, um die Knotentypen zu bestimmen, die für eine Abwärtsskalierung verfügbar sind.
 - `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 Amazon-ElastiCache-API-Referenz.

2. Führen Sie die ElastiCache-API-Aktion `ModifyRedplicationGroup` mit dem folgenden Parameter aus, um die derzeitige Replikationsgruppe auf den neuen Knotentyp zu skalieren.
 - `ReplicationGroupId` der Name der Replikationsgruppe.
 - `CacheNodeType` Der neue, kleinere Knotentyp des Cache-Clusters in dieser Replikationsgruppe. Der Wert muss einer der Instance-Typen sein, die in Schritt 1 mit dem Aufruf `ListAllowedNodeTypeModifications` 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 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 Amazon-ElastiCache-API-Referenz.

3. Wenn Sie `ApplyImmediately=true` verwenden, überwachen Sie mit der ElastiCache-API-Aktion `DescribeReplicationGroups` mit den folgenden Parametern den Status der Replikationsgruppe. 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 Amazon-ElastiCache-API-Referenz.

Erhöhen der Lesekapazität

Wenn Sie die Lesekapazität erhöhen möchten, fügen Sie Read Replicas (bis zu fünf) zur Redis-Replikationsgruppe hinzu.

Sie können die Lesekapazität eines Redis-Clusters mit der ElastiCache-Konsole, dem AWS CLI oder der ElastiCache-API skalieren. Weitere Informationen finden Sie unter [Hinzufügen eines Lesereplikats für Redis-Replikationsgruppen \(Clustermodus deaktiviert\)](#).

Verringern der Lesekapazität

Um die Lesekapazität zu verringern, löschen Sie Read Replicas aus dem Redis-Cluster mit Replikaten (in der API/CLI als Replikationsgruppe bezeichnet). 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 eines Lesereplikats für Redis-Replikationsgruppen \(Clustermodus deaktiviert\)](#).

Skalieren von Clustern in Redis (Clustermodus aktiviert)

Wenn sich die Anforderungen an Ihre Cluster ändern, können Sie beschließen, die Leistung zu verbessern oder die Preise zu senken, indem Sie die Anzahl der Shards in Ihrem Redis-Cluster (Cluster-Modus 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: `FreeableMemorySwapUsage`, und `BytesUseForCache`.

- CPU- oder Netzwerkengpässe:

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 Metriken überwachen: `CPUUtilization`, `NetworkBytesIn` `CurrConnections`, `NetworkBytesOut` 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: `FreeableMemory`, `SwapUsageBytesUseForCache`, `CPUUtilization`, `NetworkBytesIn`, `NetworkBytesOut`, und `CurrConnectionsNewConnections`

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 Verschlechterung ist, hängt von Ihrer normalen CPU-Auslastung und Ihren Daten ab.

Es gibt zwei Möglichkeiten der Skalierung Ihres Redis-Clusters (Cluster-Modus aktiviert): 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 der Speicherkapazität des Clusters durch einskalieren oder herunterskalieren verkleinern, stellen Sie sicher, dass die neue Konfiguration über ausreichend Speicher für Ihre Daten und den Redis-Overhead verfügt.

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

Inhalt

- [Offline-Resharding und Shard-Rebalancing für Redis \(Cluster-Modus aktiviert\)](#)
- [Online-Resharding und Shard-Rebalancing für 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 entfernen \(API\) ElastiCache](#)
 - [Online-Shard-Rebalancing](#)
 - [Online-Shard-Rebalancing \(Console\)](#)
 - [Online-Shard-Rebalancing \(AWS CLI\)](#)
 - [Online-Shard-Rebalancing \(API\) ElastiCache](#)

- [Vertikales Online-Skalieren durch Ändern des Knotentyps](#)
 - [Online-aufwärtsskalieren](#)
 - [Aufwärtsskalieren eines Redis-Cache-Clusters \(Konsole\)](#)
 - [Aufwärtsskalieren eines Redis-Cache-Clusters \(AWS CLI\)](#)
 - [Skalierung von Redis-Cache-Clustern \(ElastiCache API\)](#)
 - [Online-abwärtsskalieren](#)
 - [Abwärtsskalieren eines Redis-Cache-Clusters \(Konsole\)](#)
 - [Abwärtsskalieren eines Redis-Cache-Clusters \(AWS CLI\)](#)
 - [Redis-Cache-Cluster \(ElastiCache API\) herunterskalieren](#)

Offline-Resharding und Shard-Rebalancing für Redis (Cluster-Modus 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 offline erneut skalieren, können Sie zusätzlich zur Anzahl der Shards in Ihrer Replikationsgruppe Folgendes tun:

Note

Das Offline-Resharding wird für Redis-Cluster mit aktiviertem Daten-Tiering nicht unterstützt. Weitere Informationen finden Sie unter [Daten-Tiering](#).

- 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.

So konfigurieren Sie Ihre Shards (Cluster-Modus aktiviert) -Cluster offline

1. Erstellen Sie eine manuelle Sicherung Ihres vorhandenen Redis-Clusters. 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 ermitteln](#).

Online-Resharding und Shard-Rebalancing für Redis (Clustermodus aktiviert)

Durch die Verwendung von Online-Resharding und Shard-Rebalancing mit Amazon ElastiCache for Redis Version 3.2.10 oder neuer können Sie Ihr ElastiCache for Redis (Cluster-Modus aktiviert) dynamisch und ohne Ausfallzeiten skalieren. 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:

- **Ausskalieren** – Erhöhen Sie die Lese- und Schreibkapazität durch Hinzufügen von Shards (Knotengruppen) zu Ihrem Redis-Cluster (Cluster-Modus aktiviert) (Replikationsgruppe).

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.

- **Einskalieren** – Reduzieren Sie die Lese- und Schreibkapazität und damit die Preise, indem Sie Shards aus Ihrem Redis-Cluster (Cluster-Modus aktiviert) entfernen.
- **Rebalance** — Verschieben Sie die Keyspaces zwischen den Shards in Ihrem ElastiCache for Redis-Cluster (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 für Redis Online-Resharding und Rebalancing ElastiCache die folgenden Einschränkungen:

- Diese Prozesse erfordern Redis Engine Version 3.2.10 oder neuer. Informationen zum Aktualisieren der Engine-Version finden Sie unter [Engine-Versionen und Upgrades](#).
- 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

ElastiCache Für Redis Version 7 und höher: Beim Skalieren Ihres Clusters ElastiCache werden automatisch die Funktionen, die in einem der vorhandenen Knoten geladen sind (zufällig ausgewählt), auf die neuen Knoten repliziert. Wenn Ihre Anwendung [Redis-Funktionen](#) verwendet, empfehlen wir, alle Ihre Funktionen vor dem Skalieren auf alle Shards zu laden, damit Ihr ElastiCache für Redis-Cluster nicht mit unterschiedlichen Funktionsdefinitionen auf verschiedenen Shards endet.

Weitere Informationen finden Sie unter [Online-Größenanpassung von Clustern](#).

Sie können Ihre ElastiCache For-Redis-Cluster (Clustermodus aktiviert) mithilfe der API, der und der API horizontal skalieren oder neu AWS Management Console ausbalancieren. AWS CLI ElastiCache

Hinzufügen von Shards mit Online-Resharding

Sie können Ihrem Redis-Cluster (Clustermodus aktiviert) mithilfe der API AWS Management Console, AWS CLI oder Shards hinzufügen. ElastiCache Wenn Sie einem Redis-Cluster (Cluster-Modus aktiviert) Shards hinzufügen, werden alle Tags der vorhandenen Shards auf die neuen Shards übertragen.

Hinzufügen von Shards (Konsole)

Sie können den verwenden AWS Management Console , um Ihrem Redis-Cluster (Clustermodus aktiviert) einen oder mehrere Shards hinzuzufügen. Das folgende Verfahren beschreibt den Prozess.

So fügen Sie Shards zu Ihrem Redis-Cluster (Cluster-Modus aktiviert) hinzu

1. [Öffnen Sie die ElastiCache Konsole unter https://console.aws.amazon.com/elasticache/.](https://console.aws.amazon.com/elasticache/)
2. Wählen Sie im Navigationsbereich Redis clusters (Redis-Cluster) aus.
3. Suchen Sie den Namen des Redis-Clusters (Cluster-Modus aktiviert), dem Sie Shards hinzufügen möchten, und wählen Sie ihn aus (nicht das Feld links neben dem Clusternamen).

Tip

Redis (Cluster-Modus aktiviert)Clustered RedisimModecolumn

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 Redis-Cluster (Cluster-Modus aktiviert) neu konfigurieren, indem Sie Shards mit dem Befehl AWS CLI hinzufügen.

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.

Das folgende Beispiel rekonfiguriert die Keyspaces über vier Shards im Redis-Cluster `my-cluster` (Cluster-Modus aktiviert). 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" \  
    "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 in der Dokumentation unter [modify-replication-group-shard-configuration](#). AWS CLI

Hinzufügen ElastiCache von Shards (API)

Sie können die ElastiCache API verwenden, um die Shards in Ihrem Redis-Cluster (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 Redis-Cluster (Cluster-Modus aktiviert) neu konfigurieren, indem Sie Shards mithilfe der API hinzufügen. ElastiCache

Example - Shards hinzufügen

Im folgenden Beispiel werden dem Redis-Cluster `my-cluster` (Cluster-Modus aktiviert) Knotengruppen hinzugefügt, 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 unter [ModifyReplicationGroupShardConfiguration](#) in der API-Referenz. ElastiCache

Entfernen von Shards mit Online-Resharding

Sie können Shards mithilfe der API, oder ElastiCache aus Ihrem Redis-Cluster (Clustermodus aktiviert) entfernen. AWS Management Console AWS CLI

Themen

- [Entfernen von Shards \(Konsole\)](#)
- [Entfernen von Shards \(AWS CLI\)](#)
- [Shards entfernen \(API\) ElastiCache](#)

Entfernen von Shards (Konsole)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Redis-Cluster (Cluster-Modus aktiviert) neu konfigurieren können, indem Sie Shards mit der AWS Management Console entfernen.

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, ElastiCache stellen Sie sicher, 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 Redis-Cluster (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.

So entfernen Sie Shards aus Ihrem Redis-Cluster (Cluster-Modus aktiviert)

1. [Öffnen Sie die ElastiCache Konsole unter https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).
2. Wählen Sie im Navigationsbereich Redis clusters (Redis-Cluster) aus.
3. Suchen Sie den Namen des Redis-Clusters (Cluster-Modus aktiviert), aus dem Sie Shards entfernen möchten, und wählen Sie ihn aus (nicht das Feld links neben dem Clusternamen).


 Tip

-Cluster (Cluster-Modus aktiviert) haben im -Cluster den Wert 1 oder größerShardscolumn.

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 Redis-Cluster (Cluster-Modus aktiviert) neu konfigurieren können, indem Sie Shards mit der AWS CLI entfernen.

 Important

Bevor Sie Knotengruppen (Shards) aus Ihrer Replikationsgruppe entfernen, ElastiCache stellen Sie sicher, 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 Redis-Cluster (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 der Shard-IDs (Knotengruppen), die aus der Replikationsgruppe entfernt 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 Redis-Cluster `my-cluster` (Cluster-Modus aktiviert) entfernt, so dass 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 entfernen (API) ElastiCache

Sie können die ElastiCache API verwenden, um die Shards in Ihrem Redis-Cluster (Clustermodus aktiviert) online neu zu konfigurieren, indem Sie den Vorgang verwenden.

`ModifyReplicationGroupShardConfiguration`

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Redis-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie Shards mithilfe der API entfernen. ElastiCache

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 die ElastiCache API verwenden, um einen oder mehrere Shards aus Ihrem Redis-Cluster (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 der Shard-IDs (Knotengruppen), die aus der Replikationsgruppe entfernt 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 Redis-Cluster `my-cluster` (Cluster-Modus aktiviert) entfernt, so dass 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 Redis-Cluster (Clustermodus aktiviert) mithilfe der AWS Management Console API, oder neu verteilen. AWS CLI ElastiCache

Themen

- [Online-Shard-Rebalancing \(Console\)](#)
- [Online-Shard-Rebalancing \(AWS CLI\)](#)
- [Online-Shard-Rebalancing \(API\) ElastiCache](#)

Online-Shard-Rebalancing (Console)

Der folgende Prozess beschreibt, wie Sie die Shards in Ihrem Redis-Cluster (Cluster-Modus aktiviert) neu konfigurieren, indem Sie die Shards mit der AWS Management Console ausgleichen.

So gleichen Sie die Keyspaces unter den Shards in Ihrem Redis-Cluster (Cluster-Modus aktiviert) neu aus

1. Öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis clusters (Redis-Cluster) aus.

3. Wählen Sie den Namen (nicht das Feld links neben dem Namen) des Redis-Clusters (Cluster-Modus aktiviert), den Sie neu ausbalancieren möchten.

 Tip

-Cluster (Cluster-Modus aktiviert) haben im -Cluster den Wert 1 oder größerShardscolumn.

4. Wählen Sie Rebalance aus.
5. Wählen Sie nach Aufforderung die Option Rebalance aus. Möglicherweise wird eine Meldung angezeigt, die dieser ähnelt: *Slots in der Replikationsgruppe sind gleichmäßig verteilt. Nichts zu tun. (Dienst: AmazonElastiCache; Statuscode: 400; Fehlercode: InvalidReplicationGroupState; Anforderungs-ID: 2246cebd-9721-11e7-8d5b-e1b0f086c8cf)*. Wählen Sie Cancel wenn Sie das tun.

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 Redis-Cluster (Cluster-Modus aktiviert) neu konfigurieren, indem Sie die Shards mit der AWS CLI ausgleichen.

Example - Ausgleichen der Shards in einem Cluster

Das folgende Beispiel gleicht die Slots im Redis-Cluster `my-cluster` (Cluster-Modus aktiviert) so aus, dass die Slots so gleichmäßig wie möglich verteilt sind. 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 (API) ElastiCache

Sie können die ElastiCache API verwenden, um die Shards in Ihrem Redis-Cluster (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 Redis-Cluster (Clustermodus aktiviert) neu konfigurieren, indem Sie die Shards mithilfe der API neu verteilen. ElastiCache

Example - Cluster-Ausgleich

Das folgende Beispiel gleicht die Slots im Redis-Cluster `my-cluster` (Cluster-Modus aktiviert) so aus, dass die Slots so gleichmäßig wie möglich verteilt sind. 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 Amazon ElastiCache for Redis Version 3.2.10 oder neuer können Sie Ihre Redis-Cluster dynamisch mit minimalen Ausfallzeiten skalieren. Dadurch können Ihre Redis-Cluster während der Skalierung weiterhin Anfragen 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](#).

Sie haben die folgenden Möglichkeiten:

- Hochskalierung – Erhöhen Sie die Lese- und Schreibkapazität, indem Sie den Knotentyp Ihres Redis-Clusters auf die Verwendung eines größeren Knotens anpassen.

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 die Elastic Network-Schnittstelle (ENI) über ausreichend Kapazität verfügt. 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 [Verwalten von reserviertem Speicher](#).

- 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](#)
 - [Aufwärtsskalieren eines Redis-Cache-Clusters \(Konsole\)](#)
 - [Aufwärtsskalieren eines Redis-Cache-Clusters \(AWS CLI\)](#)
 - [Skalierung von Redis-Cache-Clustern \(ElastiCache API\)](#)
- [Online-abwärtsskalieren](#)
 - [Abwärtsskalieren eines Redis-Cache-Clusters \(Konsole\)](#)
 - [Abwärtsskalieren eines Redis-Cache-Clusters \(AWS CLI\)](#)
 - [Redis-Cache-Cluster \(ElastiCache API\) herunterskalieren](#)

Online-aufwärtsskalieren

Themen

- [Aufwärtsskalieren eines Redis-Cache-Clusters \(Konsole\)](#)
- [Aufwärtsskalieren eines Redis-Cache-Clusters \(AWS CLI\)](#)

- [Skalierung von Redis-Cache-Clustern \(ElastiCache API\)](#)

Aufwärtsskalieren eines Redis-Cache-Clusters (Konsole)

Das folgende Verfahren beschreibt, wie Sie einen Redis-Cluster mithilfe der ElastiCache Management Console skalieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

So führen Sie eine Aufwärtsskalierung eines Redis-Clusters durch (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 Redis clusters (Redis-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.

Aufwärtsskalieren eines Redis-Cache-Clusters (AWS CLI)

Im folgenden Verfahren wird beschrieben, wie Sie einen Redis-Cache-Cluster über die AWS CLI aufwärtsskalieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

So führen Sie eine Aufwärtsskalierung eines Redis-Cache-Clusters durch (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 in etwa folgendermaßen aus (JSON format).

```
{  
  "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-modifications](#) 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 in etwa folgendermaßen aus (JSON format).

```
{
  "ReplicationGroup": {
    "Status": "modifying",
    "Description": "my-redis-cluster",
    "NodeGroups": [
      {
        "Status": "modifying",
        "Slots": "0-16383",
        "NodeGroupId": "0001",
        "NodeGroupMembers": [
          {
            "PreferredAvailabilityZone": "us-east-1f",
            "CacheNodeId": "0001",
            "CacheClusterId": "my-redis-cluster-0001-001"
          },
          {
            "PreferredAvailabilityZone": "us-east-1d",
            "CacheNodeId": "0001",
            "CacheClusterId": "my-redis-cluster-0001-002"
          }
        ]
      }
    ],
    "ConfigurationEndpoint": {
      "Port": 6379,
      "Address": "my-redis-cluster.r7gdfi.clustercfg.us1.cache.amazonaws.com"
    },
    "ClusterEnabled": true,
    "ReplicationGroupId": "my-redis-cluster",
    "SnapshotRetentionLimit": 1,
    "AutomaticFailover": "enabled",
    "SnapshotWindow": "07:30-08:30",
    "MemberClusters": [
      "my-redis-cluster-0001-001",
      "my-redis-cluster-0001-002"
    ],
    "CacheNodeType": "cache.m3.xlarge",
    "DataTiering": "disabled",
    "PendingModifiedValues": {}
  }
}
```



```
}
```

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 Redis-Cache-Clustern (ElastiCache API)

Der folgende Prozess skaliert Ihren Cache-Cluster mithilfe der ElastiCache API von seinem aktuellen Knotentyp auf einen neuen, größeren Knotentyp. Während dieses Vorgangs aktualisiert ElastiCache für Redis die DNS-Einträge, sodass sie auf die neuen Knoten verweisen. Daher müssen Sie die Endpunkte in Ihrer Anwendung nicht aktualisieren. Bei Redis 5.0 und höher können Sie zum automatischen Failover fähige Cluster skalieren, während der Cluster online bleibt und eingehende Anforderungen ausführt. Bei Version 4.0.10 und niedriger stellen Sie bei vorherigen Versionen möglicherweise eine kurze Unterbrechung der Lese- und Schreibvorgänge vom Primärknoten fest, während der DNS-Eintrag aktualisiert wird.

Wie lange die Aufwärtsskalierung auf einen größeren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

Um einen Redis-Cache-Cluster (ElastiCache API) hochzuskalieren

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache `ListAllowedNodeTypeModifications` API-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 [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-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. Der Wert muss einer der Instance-Typen sein, die in Schritt 1 mit dem Aufruf `ListAllowedNodeTypeModifications` 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 [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

3. Falls Sie dies verwendet haben `ApplyImmediately=true`, überwachen Sie den Status der Replikationsgruppe mithilfe der ElastiCache `DescribeReplicationGroups` API-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 [DescribeReplicationGroups](#) in der Amazon ElastiCache API-Referenz.

Online-abwärtsskalieren

Themen

- [Abwärtsskalieren eines Redis-Cache-Clusters \(Konsole\)](#)
- [Abwärtsskalieren eines Redis-Cache-Clusters \(AWS CLI\)](#)
- [Redis-Cache-Cluster \(ElastiCache API\) herunterskalieren](#)

Abwärtsskalieren eines Redis-Cache-Clusters (Konsole)

Das folgende Verfahren beschreibt, wie Sie einen Redis-Cluster mithilfe der ElastiCache Management Console herunterskalieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

So führen Sie eine Abwärtsskalierung eines Redis-Clusters durch (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 Redis clusters (Redis-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.

Abwärtsskalieren eines Redis-Cache-Clusters (AWS CLI)

Im folgenden Verfahren wird beschrieben, wie Sie einen Redis-Cache-Cluster über die AWS CLI abwärtsskalieren. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

So führen Sie eine Abwärtsskalierung eines Redis-Cache-Clusters durch (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 in etwa folgendermaßen aus (JSON format).

```
{  
  "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-modifications](#) 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 in etwa folgendermaßen aus (JSON format).

```
{  
  "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.

Redis-Cache-Cluster (ElastiCache API) herunterskalieren

Der folgende Prozess skaliert Ihre Replikationsgruppe mithilfe der ElastiCache API von ihrem aktuellen Knotentyp auf einen neuen, kleineren Knotentyp. Während dieses Prozesses bedient Ihr Redis-Cluster weiterhin Anfragen mit minimalen Ausfallzeiten.

Wie lange die Abwärtsskalierung auf einen kleineren Knotentyp dauert, hängt vom Knotentyp und der Datenmenge in Ihrem derzeitigen Cache-Cluster ab.

Herunterskalierung (ElastiCache API)

1. Ermitteln Sie, auf welche Knotentypen Sie mithilfe der ElastiCache `ListAllowedNodeTypeModifications` API-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 [ListAllowedNodeTypeModifications](#) in der Amazon ElastiCache API-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. Der Wert muss einer der Instance-Typen sein, die in Schritt 1 mit dem Aufruf `ListAllowedNodeTypeModifications` 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 [ModifyReplicationGroup](#) in der Amazon ElastiCache API-Referenz.

Erste Schritte mit JSON in ElastiCache for Redis

ElastiCache for Redis unterstützt das native JSON-Format (JavaScript Object Notation), welches eine einfache, schemalose Methode zum Codieren komplexer Datensätze innerhalb von Redis-Clustern ist. Sie können Daten nativ mithilfe des JSON-Formats (JavaScript Object Notation) in Redis-Clustern speichern, und darauf zugreifen. Sie können die in diesen Clustern gespeicherten JSON-Daten aktualisieren, ohne benutzerdefinierten Code verwalten zu müssen, um ihn zu serialisieren und zu deserialisieren.

Über die Verwendung von Redis-API-Operationen für Anwendungen, die über JSON arbeiten, hinaus können Sie jetzt bestimmte Teile eines JSON-Dokuments effizient abrufen und aktualisieren, ohne das gesamte Objekt bearbeiten zu müssen. Dies kann die Leistung verbessern und Kosten senken. Sie können den Inhalt Ihres JSON-Dokuments auch mit der [Goessner](#)-Abfrage abfragen.

Nachdem Sie einen Cluster mit einer unterstützten Engine-Version erstellt haben, sind der JSON-Datentyp und die zugehörigen Befehle automatisch verfügbar. Dies ist API- und RDB-kompatibel mit Version 2 des RedisJSON-Moduls, sodass Sie vorhandene JSON-basierte Redis-Anwendungen problemlos in ElastiCache für Redis migrieren können. Weitere Informationen zu den unterstützten Redis-Befehlen finden Sie unter [Unterstützte Redis-JSON-Befehle](#).

Die JSON-bezogenen Metriken `JsonBasedCmds` und `JsonBasedCmdsLatency` werden in CloudWatch integriert, um die Nutzung dieses Datentyps zu überwachen. Weitere Informationen finden Sie unter [Metriken für Redis](#).

Note

Um JSON verwenden zu können, müssen Sie die Redis-Engine-Version 6.2.6 oder höher ausführen.

Themen

- [Übersicht über Redis-JSON-Datentypen](#)
- [Unterstützte Redis-JSON-Befehle](#)

Übersicht über Redis-JSON-Datentypen

ElastiCache for Redis unterstützt eine Reihe von 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 Redis-Befehle.

Terminologie

| Begriff | Beschreibung |
|---------------|--|
| JSON-Dokument | Bezieht sich auf den Wert eines Redis-JSON-Schlüssels. |

| Begriff | Beschreibung |
|--------------|---|
| JSON-Wert | Bezieht sich auf eine Teilmenge eines JSON-Dokuments, einschließlich des Stammverzeichnisses, das das gesamte Dokument darstellt. Ein Wert könnte ein Container oder ein Eintrag innerhalb eines Containers sein. |
| JSON-Element | Äquivalent zu JSON-Wert. |

Unterstützter JSON-Standard

Das JSON-Format ist mit [RFC 7159](#) und dem [ECMA-404](#)-JSON-Datenaustauschstandard konform. UTF-8 [Unicode](#) wird im JSON-Text unterstützt.

Stammelement

Das Stammelement kann von jedem JSON-Datentyp stammen. Beachten Sie, dass in früheren RFC 4627 nur Objekte oder Arrays als Stammwerte zugelassen waren. Seit dem Update auf RFC 7159 kann das Stammverzeichnis eines JSON-Dokuments einen beliebigen JSON-Datentyp haben.

Begrenzung der Dokumentgröße

JSON-Dokumente werden intern in einem Format gespeichert, das für schnellen Zugriff und Änderung 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, was der Größe der In-Memory-Datenstruktur entspricht, nicht der JSON-Zeichenfolge. Sie können mit dem `JSON.DEBUG MEMORY`-Befehl den Speicherplatz überprüfen, der von einem JSON-Dokument verbraucht wird.

JSON-ACLs

- Ähnlich wie bei den vorhandenen Pro-Datentyp-Kategorien (`@string`, `@hash` usw.) wird eine neue Kategorie `@json` hinzugefügt, um die Verwaltung des Zugriffs auf JSON-Befehle und -Daten zu vereinfachen. Keine anderen vorhandenen Redis-Befehle sind Mitglieder der Kategorie `@json`. Alle JSON-Befehle erzwingen alle Keyspace- oder Befehlseinschränkungen und -berechtigungen.

- Es gibt fünf bestehende Redis-ACL-Kategorien, die um die neuen JSON-Befehle ergänzt wurden: @read, @write, @fast, @slow und @admin. Die folgende Tabelle zeigt die Zuordnung von JSON-Befehlen zu den entsprechenden Kategorien an.

ACL

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

| JSON-Befehl | @read | @write | @fast | @slow | @admin |
|--------------------|-------|--------|-------|-------|--------|
| JSON.NUMI NCRBY | | y | y | | |
| JSON.NUMM ULTBY | | y | y | | |
| JSON.OBJK EYS | y | | y | | |
| JSON.OBJL EN | y | | y | | |
| JSON.RESP | y | | y | | |
| JSON.SET | | y | | y | |
| JSON.STRA PPEND | | y | y | | |
| 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 Array 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 Arrays

„verschachtelt“ wird. 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 benötigen einen Redis-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 JsonPath-Syntax, die von [Goessner](#) beschrieben wird, wie in der folgenden Tabelle gezeigt. 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. |
| () | 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. |

| Symbol/Ausdruck | Beschreibung |
|-----------------|--|
| && | Logisches UND; wird verwendet, um mehrere Filterausdrücke zu kombinieren. |
| | Logisches ODER; wird verwendet, um mehrere Filterausdrücke zu kombinieren. |

Beispiele

Die folgenden Beispiele bauen auf den Beispiel-XML-Daten von [Goessner](#) auf, die wir durch Hinzufügen zusätzlicher Felder geändert 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 |
| <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 JSON-Struktu. |
| <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 inklusiv). |
| <code>\$.book[0:4:2]</code> | Bücher bei Index 0, 2. |

| Pfad | Beschreibung |
|--|---|
| <code>\$.book[?(@.isbn)]</code> | Alle Bücher mit einer ISBN-Nummer. |
| <code>\$.book[?(@.price<10)]</code> | Alle Bücher günstiger als 10 USD. |
| <code>'\$.book[?(@.price < 10)]'</code> | Alle Bücher günstiger als 10 USD. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.) |
| <code>'\$.book[?(@["price"] < 10)]'</code> | Alle Bücher günstiger als 10 USD. |
| <code>'\$.book[?(@.["price"] < 10)]'</code> | Alle Bücher günstiger als 10 USD. |
| <code>\$.book[?(@.price>=10&&@.price<=100)]</code> | Alle Bücher im Preisbereich von 10 bis 100 USD inklusive. |
| <code>'\$.book[?(@.price>=10 && @.price<=100)]'</code> | Alle Bücher im Preisbereich von 10 bis 100 USD inklusive. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.) |
| <code>\$.book[?(@.sold==true @.in-stock==false)]</code> | Alle Bücher verkauft oder ausverkauft. |
| <code>'\$.book[?(@.sold == true @.in-stock == false)]'</code> | Alle Bücher verkauft oder ausverkauft. (Der Pfad muss in Anführungszeichen gesetzt werden, wenn er Leerzeichen enthält.) |
| <code>'\$.store.book[?(@.["category"] == "fiction")]'</code> | Alle Bücher der Kategorie Belletristik. |
| <code>'\$.store.book[?(@.["category"] != "fiction")]'</code> | 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. |

| Pfad | Beschreibung |
|---|-------------------------------|
| <code>["store"]["book"][0]["title"]</code> | Der Titel des ersten Buches. |
| <code>["store"]["book"][-1]["title"]</code> | 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. |

JSON-verwandte Metriken

Die folgenden JSON-Infometriken werden bereitgestellt:

| Informationen | Beschreibung |
|-------------------------|---|
| json_total_memory_bytes | JSON-Objekten zugewiesener Gesamtspeicher |
| json_num_documents | Gesamtzahl von Dokumenten in Redis |

Führen Sie zum Abfragen von Kernmetriken den folgenden Redis-Befehl aus:

```
info json_core_metrics
```

Wie ElastiCache for Redis mit JSON interagiert

Im folgenden Abschnitt wird beschrieben, wie ElastiCache for Redis mit dem JSON-Datentyp interagiert.

Rangfolge der Operatoren

Bei der Bewertung bedingter Ausdrücke zum Filtern, haben `&&` 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

Die maximale Pfad-Verschachtelungsgrenze in ElastiCache für Redis ist 128. Ein Wert wie `$.a.b.c.d...` kann also nur 128 Level erreichen.

Umgang mit numerischen Werten

JSON hat keine separaten Datentypen für ganze Zahlen und Gleitkommazahlen. Sie werden alle Zahlen genannt.

Numerische Repräsentationen:

Wenn eine JSON-Nummer bei der Eingabe empfangen wird, wird sie in eine der beiden internen Binärdarstellungen umgewandelt: eine 64-Bit-Ganzzahl oder eine doppelt genaue 64-Bit-Gleitkommazahl. Die Ursprüngliche Zeichenfolge und alle ihre Formatierungen werden nicht beibehalten. Wenn also eine Zahl als Teil einer JSON-Antwort ausgegeben wird, wird sie von 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, ergibt sich daraus automatisch eine doppelt genaue 64-Bit-Gleitkommazahl.
- Wenn mindestens eine der Zahlen eine Gleitkommazahl ist, ergibt sich daraus eine doppelt genaue 64-Bit-Gleitkommazahl.
- Wenn das Ergebnis den Bereich einer doppelt genauen 64-Bit-Gleitkommazahl überschreitet, gibt der Befehl einen `OVERFLOW`-Fehler aus.

Eine detaillierte Liste der verfügbaren Befehle finden Sie unter [Unterstützte Redis-JSON-Befehle](#).

Direktes Array-Filtern

ElastiCache for Redis filtert Array-Objekte direkt.

Für Daten wie `[0, 1, 2, 3, 4, 5, 6]` und eine Pfadabfrage wie `$[?(@<4)]` oder Daten wie `{"my_key": [0, 1, 2, 3, 4, 5, 6]}` und eine Pfadabfrage wie `$.my_key[?(@<4)]` würde ElastiCache for Redis unter beiden Umständen `[1,2,3]` zurückgeben.

Array-Indizierung

ElastiCache for Redis ermöglicht sowohl positive als auch negative Indizes für Arrays. 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, rundet ElastiCache for Redis Array-Indizes nicht ab oder auf. Wenn Sie also ein Array mit einer Länge von 5, einen Calling-Index von 5 oder höher oder -6 oder niedriger haben, würde sich daraus kein Ergebnis ergeben.

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 Redis-JSON-Befehle

ElastiCache for Redis unterstützt die folgenden Redis JSON-Befehle:

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 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.
- `SYNTAXERR`-Fehler, wenn eines der eingegebenen JSON-Argumente 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.
- `SYNTAXERR`-Fehler, wenn eines der eingegebenen JSON-Argumente 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", "c"], ["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 skalaren JSON-Werts 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 Redis-Schlüssel des JSON-Dokumenttyps.
- `path` (erforderlich) – Ein JSON-Pfad.
- `json-scalar` (erforderlich) – Der Skalarwert für die Suche nach . JSON-Skalar 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 (exklusiv). 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 Redis-Schlüssel des JSON-Dokumenttyps.
- 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.
- OUTFOUBOUNDARIES-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.
- OUTFOUBOUNDARIES-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 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 Redis-Schlüssel des JSON-Dokumenttyps.
- **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-String, der den eingeblendeten JSON-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 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.
- OUTFBOUNDARIES-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.
- OUTFBOUNDARIES-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 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 Speicherauslastung eines JSON-Wertes 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 JSON-Werte. Jeder Containerwert, mit Ausnahme des Root-Containers, zählt als ein zusätzliches Feld.

- HELP – Druckt Hilfmeldungen des Befehls.

Syntax

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

Hängt vom Unterbefehl ab:

MEMORY

- Wenn der Pfad eine erweiterte Syntax ist:
 - Gibt ein Array von ganzen Zahlen zurück, das die Speichergröße (in Byte) des JSON-Werts bei jedem Pfad darstellt.
 - Gibt ein leeres Array zurück, wenn der Redis-Schlüssel nicht existiert.
- Wenn der Pfad eine eingeschränkte Syntax hat:
 - Gibt eine Ganzzahl, Speichergröße und den JSON-Wert in Byte zurück.
 - Gibt Null zurück, wenn der Redis-Schlüssel nicht existiert.

FIELDS

- Wenn der Pfad eine erweiterte Syntax ist:
 - Gibt ein Array von ganzen Zahlen zurück, die die Anzahl der Felder mit JSON-Wert in jedem Pfad darstellen.
 - Gibt ein leeres Array zurück, wenn der Redis-Schlüssel nicht existiert.
- Wenn der Pfad eine eingeschränkte Syntax hat:
 - Gibt eine Ganzzahl und die Anzahl der Felder des JSON-Werts zurück.
 - Gibt Null zurück, wenn der 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 der Pfad das Stammverzeichnis ist, entspricht dies dem Löschen des Schlüssels aus Redis.

Syntax

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

- key (erforderlich) – Ein Redis-Schlüssel des JSON-Dokumenttyps.
- path (optional) – Ein JSON-Pfad. Es gilt der Root-Standardwert, falls nicht angegeben.

Ergebnis

- Anzahl der gelöschten Elemente.
- 0, wenn der 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 in einem oder mehreren Pfaden zurück.

Syntax

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

- **key** (erforderlich) – Ein Redis-Schlüssel des JSON-Dokumenttyps.
- **EINRÜCKUNG/NEUE ZEILE/LEERTASTE** (optional) – Steuert das Format der zurückgegebenen JSON-Zeichenfolge „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 Legacy-Kompatibilität vorhanden sein und hat keine andere Auswirkung.
- path (optional) – Null oder mehr JSON-Pfade, standardmäßig das Stammverzeichnis, wenn keine Angabe. 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 stringiertes JSON-Objekt zurück, in 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.*
["\n21 2nd Street\n","\nNew York\n","\nNY\n","\n10021-3100\n"]
127.0.0.1:6379> JSON.GET k1 indent "\t" space " " NEWLINE "\n" $.address.*
["\n\t21 2nd Street\n","\n\tNew York\n","\n\tNY\n","\n\t10021-3100\n\n"]
```

```
127.0.0.1:6379> JSON.GET k1 $.firstName $.lastName $.age
{"$.firstName":["John"],$.lastName":["Smith"],$.age":[27]}"
127.0.0.1:6379> JSON.SET k2 . '{"a":{, "b":{"a":1}, "c":{"a":1, "b":2}}'
OK
127.0.0.1:6379> json.get k2 $.*
"[{},{\"a\":1},{\"a\":1,\"b\":2},1,1,2]"
```

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": "\t21 2nd Street",\n\t"city": "\tNew York",\n\t"state": "\tNY",\n
\t"zipcode": "\t10021-3100"}
127.0.0.1:6379> JSON.GET k1 .firstName .lastName .age
{"$.firstName":"John",$.lastName":"Smith",$.age":27}"
```

JSON.MGET

Ruft serialisierte JSONs am Pfad aus mehreren Dokumentschlüsseln ab. Es gibt null für einen nicht existierenden Schlüssel oder JSON-Pfad zurück.

Syntax

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

- key (erforderlich) – Ein oder mehrere 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 serialisierten JSON gefüllt, wie es sich im Pfad befindet, oder (b) mit null, wenn der Schlüssel nicht existiert, der Pfad nicht im Dokument festgehalten oder ungültig (Syntaxfehler) ist.
- Wenn einer der angegebenen Schlüssel existiert und kein JSON-Schlüssel ist, gibt der Befehl den Fehler `WRONGTYPE` 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 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 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 Bereichs der doppelt genauen 64-Bit-Gleitkommazahl 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 der doppelt genauen 64-Bit-Gleitkommazahl 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\":[],\"b\":{\"a\":2},\"c\":{\"a\":2,\"b\":3},\"d\":{\"a\":2,\"b\":3,\"c\":4}}"

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

```

"[null,2]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.c.* 1
"[null,null]"
127.0.0.1:6379> JSON.NUMINCRBY k3 $.d.* 1
"[2,null,4]"
127.0.0.1:6379> JSON.GET k3
"{\"a\":{\"a\":{\"a\":\"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 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 eines doppelt genauen 64-Bit-IEEE-Gleitkommawerts 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 eines doppelt genauen 64-Bit-IEEE-Werts 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 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 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 Schlüssel 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 im angegebenen Pfad im Redis-Serialisierungsprotokoll (RESP) zurück. Wenn der Wert Container lautet, ist die Antwort ein RESP-Array oder verschachteltes Array.

- JSON null wird dem RESP Null Bulk String zugeordnet.
- Boolesche JSON-Werte werden den jeweiligen RESP Simple Strings zugeordnet.
- Ganzzahlen werden RESP-Ganzzahlen zugeordnet.

- Doppelt genaue 64-Bit-Gleitkommazahlen werden RESP-Bulk-Strings zugeordnet.
- JSON-Strings werden RESP Bulk Strings zugeordnet.
- JSON-Arrays werden als RESP-Arrays dargestellt, wobei das erste Element die einfache Zeichenfolge [ist, gefolgt von den Elementen des Arrays.
- JSON-Objekte werden als RESP-Arrays dargestellt, wobei das erste Element die einfache Zeichenfolge { ist, gefolgt von Schlüssel-Wert-Paaren, von denen jedes ein RESP-Bulk-String ist.

Syntax

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

- key (erforderlich) – Ein 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. 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 in einem Pfad repräsentiert.
- 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 im Pfad fest.

Wenn der Pfad ein Objektmitglied aufruft:

- Wenn das übergeordnete Element nicht vorhanden ist, gibt der Befehl den Fehler NONEXISTENT aus.
- Wenn das übergeordnete Element vorhanden aber kein Objekt ist, gibt der Befehl einen FEHLER zurück.
- 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. Ansonsten gibt der Befehl den Fehler NONEXISTENT 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 vorhanden ist, gibt der Befehl den Fehler NONEXISTENT aus.
- Wenn das übergeordnete Element vorhanden aber kein Array ist, gibt der Befehl einen FEHLER zurück.

- Wenn das übergeordnete Element vorhanden ist, der Index jedoch außerhalb der Grenzen liegt, gibt der Befehl den Fehler OUTFOUBOUNDARIES 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 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 Redis-Schlüssel des JSON-Dokumenttyps.
- path (erforderlich) – Ein JSON-Pfad. Für einen neuen Redis-Schlüssel muss der JSON-Pfad dem Stamm „.“ entsprechen.
- NX (optional) – Wenn der Pfad der Stamm ist, legen Sie den Wert nur fest, wenn der Redis-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 der Pfad der Stamm ist, legen Sie den Wert nur fest, wenn der Redis-Schlüssel existiert. 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) OUTOFBOUNDARIES Array index is out of bounds
```

JSON.STRAPPEND

Hängt eine Zeichenfolge an die JSON-Zeichenfolgen im Pfad an.

Syntax

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

- **key** (erforderlich) – Ein Redis-Schlüssel des JSON-Dokumenttyps.
- **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.
- `SYNTAXERR`-Fehler, wenn das angegebene 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.
- `WRONGTYPE`-Fehler, wenn das angegebene 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 Länge des JSON-Zeichenfolgenwerts im Pfad ab.

Syntax

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

- **key** (erforderlich) – Ein 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 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 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 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 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
```

Markieren Ihrer ElastiCache-Ressourcen

Um Sie bei der Verwaltung Ihrer Cluster und anderer ElastiCache-Ressourcen zu unterstützen, können Sie jeder Ressource eigene Metadaten in Form von Tags zuweisen. Mit Tags (Markierungen) 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 (Markierung) ist eine Markierung, die Sie einer AWS-Ressource zuordnen. Jeder Tag (Markierung) besteht aus einem Schlüssel und einem optionalen Wert, beides können Sie bestimmen. Mit Tags können Sie Ihre AWS-Ressourcen auf unterschiedliche Weise kategorisieren, beispielsweise nach Zweck oder Besitzer. Sie können beispielsweise eine Reihe von Tags für die ElastiCache-Cluster Ihres Kontos definieren, die Ihnen dabei helfen, den Besitzer und die Benutzergruppe jeder Instance zu verfolgen.

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 für ElastiCache keine semantische Bedeutung und werden streng 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 AWS Management Console, AWS CLI und die ElastiCache-API verwenden.

Wenn Sie IAM verwenden, 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 markieren, die bereits in Ihrem Konto vorhanden sind. In der Tabelle unten werden die Ressourcen aufgeführt, die das Markieren unterstützen. Wenn Sie die AWS Management Console verwenden, 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 keine semantische Bedeutung für den ElastiCache-Dienst.

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 die Amazon ElastiCache-API, die AWS-CLI oder ein AWS-SDK verwenden, können Sie den Tags-Parameter der entsprechenden ElastiCache-API-Aktion verwenden, um Tags anzuwenden. Diese sind:

- `CreateServerlessCache`
- `CreateCacheCluster`
- `CreateReplicationGroup`
- `CopyServerlessCacheSnapshot`
- `CopySnapshot`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`
- `CreateServerlessCacheSnapshot`
- `CreateSnapshot`
- `CreateUserGroup`
- `CreateUser`

- `PurchaseReservedCacheNodesOffering`

Die folgende Tabelle beschreibt die ElastiCache-Ressourcen, die mit Tags versehen werden können, und die Ressourcen, die bei der Erstellung mit der ElastiCache-API, der AWS-CLI oder einem AWS-SDK mit Tags versehen werden können.

Tagging-Unterstützung für ElastiCache-Ressourcen

| Unterstützt Tags (Markierungen) | Unterstützt Markierung bei der Erstellung |
|---------------------------------|---|
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |
| Ja | Ja |

 Note

Globale Datenspeicher können nicht mit Tags versehen werden.

Sie können in Ihren IAM-Richtlinien tagbasierte Berechtigungen auf Ressourcenebene auf die ElastiCache-API-Aktionen anwenden, die das Tagging bei der Erstellung unterstützen, um eine granulare Kontrolle über die Benutzer und Gruppen zu implementieren, 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 (Markierung)-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 zulässt, können andere Dienste dies einschränken. Folgende Zeichen sind dienstübergreifend 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 Präfix aws : ist zur Verwendung in AWS 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

- Erstellen eines Serverless-Caches mithilfe von Tags

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine redis \  
  --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 redis \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

```
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- Erstellen eines Serverless-Snapshots mit Tags.

```
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.

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. AddTagsToResource-Aktion für einen Cluster nur zulassen, wenn der Cluster das Tag „Project=XYZ“ hat.

```
{  
  "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. Verweigerung CopySnapshot-Aktion, wenn der Quell-Snapshot das Tag „Project=XYZ“ hat und das Anforderungs-Tag Service=Elasticache ist.

```

{
  "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 ElastiCache Kostenzuordnungs-Tags hinzufügen, können Sie Kosten verfolgen, indem Sie Ausgaben auf Ihren Rechnungen nach Ressourcen-Tag-Werten gruppieren.

Ein ElastiCache-Kostenzuordnungs-Tag ist ein Schlüssel-Wert-Paar, das Sie definieren und einer ElastiCache-Ressource zuordnen. Bei Schlüssel 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 Ihre eigene Kostenstruktur widerspiegelt. Dazu müssen Sie sich registrieren, um Ihre AWS-Kontorechnung mit Tag (Markierung)-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 gesamten ElastiCache-Kosten nach Regionen aufgeschlüsselt sehen. Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#) im AWS Billing-Benutzerhandbuch.

Sie können Redis-Knoten ElastiCache-Kostenzuordnungs-Tags hinzufügen. Wenn Sie ein Tag hinzufügen, auflisten, ändern, kopieren oder entfernen, wird die Operation nur auf den angegebenen Knoten angewendet.

Eigenschaften von ElastiCache-Kostenzuordnungs-Tags

- Kostenzuordnungs-Tags werden auf ElastiCache-Ressourcen angewendet, die in CLI- und API-Operationen als ARN angegeben sind. Der Ressourcentyp ist ein „Cluster“.

Beispiel-ARN: `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 ordnet Ihren Tags keine semantische Bedeutung zu. Tags werden streng als Zeichenketten interpretiert. AWS setzt nicht automatisch Tags für ElastiCache-Ressourcen.

Verwalten Ihrer Kostenzuordnungs-Tags mithilfe der AWS CLI

Zum Hinzufügen, Ändern oder Entfernen von Kostenzuordnung-Tags können Sie das AWS CLI verwenden.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Kostenzuordnungs-Tags werden auf ElastiCache-for-Redis-Knoten angewendet. Der Knoten, der gekennzeichnet werden soll, wird mit einem ARN (Amazon Resource Name) angegeben.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Themen

- [Auflisten von Tags mithilfe der AWS CLI](#)
- [Hinzufügen von Tags mithilfe der AWS CLI](#)
- [Ändern von Tags mithilfe der AWS CLI](#)
- [Entfernen von Tags mithilfe der AWS CLI](#)

Auflisten von Tags mithilfe der AWS CLI

Sie können die AWS CLI verwenden, um Tags für eine vorhandene ElastiCache-Ressource aufzulisten, indem Sie die [list-tags-for-resource](#)-Operation anwenden.

Der folgende Code listet mithilfe der AWS CLI die Tags für den Redis-Knoten `my-cluster-001` im `my-cluster`-Cluster in der Region `us-west-2` auf.

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 für die Ressource keine Tags vorhanden sind, wird eine leere TagList ausgegeben.

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

Weitere Informationen finden Sie unter AWS CLI für ElastiCache [list-tags-for-resource](#).

Hinzufügen von Tags mithilfe der AWS CLI

Sie können AWS CLI verwenden, um Tags zu einer vorhandenen ElastiCache-Ressource hinzuzufügen, indem Sie den [add-tags-to-resource](#)-CLI-Vorgang verwenden. 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 fügt mit der AWS CLI die Schlüssel `Service` und `Region` mit den Werten `elasticache` bzw. `us-west-2` dem Knoten `my-cluster-001` im Cluster `my-cluster` in der Region „us-west-2“ hinzu.

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 unter AWS CLI für ElastiCache [add-tags-to-resource](#).

Sie können auch mithilfe der AWS CLI Tags einem Cluster hinzufügen, wenn Sie mit der Operation [create-cache-cluster](#) einen neuen Cluster erstellen. Sie können beim Erstellen eines Clusters über

die ElastiCache-Managementkonsole keine Tags hinzufügen. Nachdem der Cluster erstellt wurde, können Sie über die Konsole Tags zum Cluster hinzufügen.

Ändern von Tags mithilfe der AWS CLI

Sie können mithilfe der AWS CLI Tags auf einem Knoten in einem ElastiCache-for-Redis-Cluster ändern.

Ändern Sie Tags wie folgt:

- Verwenden Sie [add-tags-to-resource](#) zum Hinzufügen eines neuen Tags und Wertes oder zum Ändern des zu einem vorhandenen Tag gehörenden Wertes.
- Verwenden Sie [remove-tags-from-resource](#) zum Entfernen angegebener Tags aus der Ressource.

zum Entfernen von Tags aus dem Cluster.

Entfernen von Tags mithilfe der AWS CLI

Sie können mithilfe der AWS CLI Tags aus einem vorhandenen Knoten eines ElastiCache-Clusters entfernen, indem Sie die Operation [remove-tags-from-resource](#) ausführen.

Der folgende Code entfernt mithilfe der AWS CLI die Tags mit den Schlüsseln `Service` und `Region` aus dem Knoten `my-cluster-001` im Cluster `my-cluster` in der Region „us-west-2“.

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 unter AWS CLI für ElastiCache [remove-tags-from-resource](#).

Verwalten Sie Ihre Kostenzuordnungs-Tags mit der ElastiCache-API

Sie können die ElastiCache-API verwenden, um Kostenzuordnungs-Tags hinzuzufügen, zu ändern oder zu entfernen.

Kostenzuordnungs-Tags werden auf ElastiCache-for-Memcached-Cluster angewendet. Der Cluster, der gekennzeichnet werden soll, wird mit einem ARN (Amazon Resource Name) angegeben.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Themen

- [Auflisten von Tags mit der ElastiCache-API](#)
- [Hinzufügen von Tags mithilfe der ElastiCache-API](#)
- [Ändern von Tags mithilfe der ElastiCache-API](#)
- [Entfernen von Tags mithilfe der ElastiCache-API](#)

Auflisten von Tags mit der ElastiCache-API

Sie können die ElastiCache-API verwenden, um Tags für eine vorhandene Ressource mithilfe des Vorgangs [ListTagsForResource](#) aufzulisten.

Der folgende Code verwendet die ElastiCache-API, um die Tags für die 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 mithilfe der ElastiCache-API

Sie können die ElastiCache-API verwenden, um mithilfe des Vorgangs [AddTagsToResource](#) Tags zu einem vorhandenen ElastiCache-Cluster hinzuzufügen. 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` und `us-west-2` bzw. der Ressource `my-cluster-001` in der Region `us-west-2` hinzuzufügen.

```
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 Amazon-ElastiCache-API-Referenz.

Ändern von Tags mithilfe der ElastiCache-API

Sie können die ElastiCache-API verwenden, um die Tags in einem ElastiCache-Cluster zu ändern.

Den Wert eines Tags ändern:

- Verwenden Sie die Operation [AddTagsToResource](#) zum Hinzufügen eines neuen Tags und Wertes oder zum Ändern des Wertes eines vorhandenen Tags.
- Verwenden Sie [RemoveTagsFromResource](#) zum Entfernen von Tags aus der Ressource.

Die Ausgabe beider Operationen ist eine Liste der Tags und deren Werte für die angegebene Ressource.

Verwenden Sie [RemoveTagsFromResource](#) zum Entfernen von Tags aus der Ressource.

Entfernen von Tags mithilfe der ElastiCache-API

Sie können mithilfe der ElastiCache-API Tags aus einem vorhandenen ElastiCache-for-Redis-Knoten entfernen, indem Sie die Operation [RemoveTagsFromResource](#) ausführen.

Der folgende Code entfernt mithilfe der ElastiCache-API die Tags mit den Schlüsseln `Service` und `Region` aus dem Knoten `my-cluster-001` im Cluster `my-cluster` in der Region `us-west-2`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RemoveTagsFromResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster-001  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&TagKeys.member.1=Service  
&TagKeys.member.2=Region  
&Version=2015-02-02  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

Verwendung von Amazon ElastiCache Well-Architected Lens

Dieser Abschnitt beschreibt Amazon DynamoDB Well-Architected Lens, eine Sammlung von Designprinzipien und Leitlinien für die Gestaltung gut strukturierter ElastiCache-Workloads.

- ElastiCache Lens ist eine Ergänzung zum [AWS Well-Architected Framework](#).
- Jede Säule enthält eine Reihe von Fragen, die helfen sollen, die Diskussion rund um eine ElastiCache-Architektur in Gang zu bringen.
 - 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.

- [Workload](#) – eine Reihe von Komponenten, die zusammen einen Geschäftswert bieten. Beispiele für Workloads sind Marketing-Websites, E-Commerce-Websites, das Backend für eine mobile App, Analyseplattformen usw.

Themen

- [Säule „Operational Excellence“ von Amazon ElastiCache Well Architected Lens](#)
- [Säule der Sicherheit von Amazon ElastiCache Well-Architected Lens](#)
- [Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens](#)
- [Amazon ElastiCache – Well-Architected Lens – Säule der Leistungseffizienz](#)
- [Säule der Kostenoptimierung von Amazon ElastiCache 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 ElastiCache-Cluster?](#)
- [OE 3: Wie verwalten Sie Ihre ElastiCache-Cluster-Ressourcen und halten Ihren Cluster auf dem neuesten Stand?](#)
- [OE 4: Wie verwalten Sie die Verbindungen der Clients mit Ihren ElastiCache-Clustern?](#)
- [OE 5: Wie stellen Sie ElastiCache-Komponenten für einen Workload bereit?](#)
- [OE 6: Wie planen und minimieren Sie Ausfälle?](#)
- [OE 7: Wie beheben Sie Probleme mit Redis-Engine-Ereignissen?](#)

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, wie ein Failover, einen Knotenwechsel, 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 von ElastiCache generierten Ereignisse in der ElastiCache-Konsole (nachdem Sie Ihre Region ausgewählt haben) oder verwenden Sie den Befehl [describe-events](#) von [Amazon Command Line Interface](#) (AWS CLI) und die [ElastiCache-API](#). Konfigurieren Sie ElastiCache so, dass Benachrichtigungen für wichtige Cluster-Ereignisse mit Amazon Simple Notification Service (Amazon SNS) gesendet werden. Wenn Sie Amazon SNS mit Ihren Clustern verwenden, können Sie programmgesteuert Aktionen bei ElastiCache-Ereignissen ausführen.
- 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, Erstellung eines Snapshots, Änderung der Cluster-Parameter, Verlängerung des CA-Zertifikats, Fehlerereignisse (Fehler bei der Cluster-Bereitstellung – VPC oder ENI, Skalierungsfehler – ENI- und Snapshot-Fehler). 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 Redis)
- [Ressourcen]:
 - [Verwalten von ElastiCache-Amazon-SNS-Benachrichtigungen](#)

- [Ereignisbenachrichtigungen und Amazon SNS](#)

- [Am besten] Nutzen Sie AWS-Produkt- und -Servicefunktionen wie SNS und Lambda-Funktionen, 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 Amazon-CloudWatch-Metriken verwenden, um Ihre Cluster zu überwachen.

[Ressourcen]: [Überwachen Sie die Lesereplikat-Endpunkte von Amazon ElastiCache für Redis \(Clustermodus deaktiviert\) mithilfe von AWS Lambda, Amazon Route 53 und Amazon SNS](#) für einen Anwendungsfall, der Lambda und SNS verwendet.

OE 2: Wann und wie skalieren Sie Ihre vorhandenen ElastiCache-Cluster?

Einführung auf Fragenebene: Die richtige Größe Ihres ElastiCache-Clusters festzulegen, ist ein Balanceakt, der bei jeder Änderung der zugrundeliegenden Workload-Typen geprüft 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. ElastiCache kann in zwei Dimensionen skaliert werden, um die Über- oder Unterauslastung Ihrer Ressourcen zu beheben. Sie können vertikal skalieren, indem Sie die Knotenkapazität erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie Knoten hinzufügen und entfernen.

- [Erforderlich] Die Überauslastung der CPU und des Netzwerks auf den Primärknoten sollte dadurch behoben werden, dass die Lesevorgänge ausgelagert und an die Replikatknoten umgeleitet werden. Verwenden Sie Replikatknoten für Lesevorgänge, um die Auslastung des Primärknotens zu reduzieren. Dies kann in Ihrer Redis-Clientbibliothek konfiguriert werden, indem Sie eine Verbindung zum ElastiCache-Reader-Endpunkt herstellen, wenn der Clustermodus deaktiviert ist, oder indem Sie den Redis-Befehl READONLY verwenden, wenn der Clustermodus aktiviert ist.

[Ressourcen]:

- [Verbindungsendpunkte ermitteln](#)
- [Richtige Clustergröße](#)
- [Redis-Befehl READONLY](#)

- [Erforderlich] Überwachen Sie die Auslastung 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 von ElastiCache für Redis 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 bewährter Methoden mit Amazon ElastiCache für Redis unter Verwendung von Amazon CloudWatch](#)
- [Skalieren von Clustern von ElastiCache für Redis](#)
- [Skalieren von Clustern von ElastiCache für Memcached](#)
- [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-Metriken über längere Zeiträume sollten in Ihre Prognose zur Auslastung der Cluster-Ressourcen einfließen. CloudWatch-Datenpunkte und -Metriken sind bis zu 455 Tage verfügbar.

[Ressourcen]:

- [Überwachung von ElastiCache für Redis mit CloudWatch-Metriken](#)
- [Überwachung von Memcached mit CloudWatch-Metriken](#)
- [Überwachung bewährter Methoden mit Amazon ElastiCache für Redis unter Verwendung von Amazon CloudWatch](#)
- [Am besten] Wenn Ihre ElastiCache-Ressourcen mit CloudFormation erstellt wurden, empfiehlt es sich, Änderungen mithilfe von CloudFormation-Vorlagen vorzunehmen. Dadurch können Sie die betriebliche Konsistenz wahren und nicht verwaltete Konfigurationsänderungen und Stackverschiebungen vermeiden.

[Ressourcen]:

- [ElastiCache-Ressourcentypreferenz für CloudFormation](#)
- [Am besten] Automatisieren Sie Ihre Skalierungsvorgänge mithilfe von Cluster-Betriebsdaten und definieren Sie Schwellenwerte in CloudWatch, um Alarme einzurichten. Verwenden Sie

CloudWatch Events und Simple Notification Service (SNS), um Lambda-Funktionen auszulösen und eine ElastiCache-API auszuführen, um Ihre Cluster automatisch 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]:

- [Verwenden von Amazon CloudWatch Alarms](#)
- [Was ist Amazon CloudWatch Alarms?](#)
- [Verwenden von AWS Lambda mit Amazon Simple Notification Service](#)
- [ElastiCache-API-Referenz](#)

OE 3: Wie verwalten Sie Ihre ElastiCache-Cluster-Ressourcen und halten Ihren Cluster auf dem neuesten Stand?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, ist es wichtig, dass Sie in der Lage sind, alle Ihre ElastiCache-Ressourcen genau zu lokalisieren und zu identifizieren. Bei der Einführung neuer Anwendungsfunktionen müssen Sie eine Symmetrie der Clusterversionen für alle Ihre ElastiCache-Umgebungstypen herstellen: Entwicklung, Test 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 der nativen ElastiCache-Funktionen kann sich Ihr Engineering-Team auf die Erfüllung der Geschäftsziele statt auf die ElastiCache-Wartung konzentrieren.

- [Am besten] Verwenden Sie die neueste verfügbare Engine-Version und wenden Sie die Self-Service-Updates an, sobald sie verfügbar sind. ElastiCache aktualisiert die zugrundeliegende Infrastruktur automatisch während des von Ihnen 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 in Amazon ElastiCache](#)
- [Hilfeseite für verwaltete Wartung und Service-Updates von Amazon ElastiCache](#)
- [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-Ressourcentypreferenz für CloudFormation](#)
- [Parametergruppen](#)

OE 4: Wie verwalten Sie die Verbindungen der Clients mit Ihren ElastiCache-Clustern?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, müssen Sie verstehen, wie sich Ihre Clients mit dem ElastiCache-Cluster verbinden, um die betrieblichen Aspekte Ihrer Anwendung (wie 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 beim Trennen der Schreibvorgänge von den Lesevorgängen die Fähigkeit verlieren, einen Schlüssel unmittelbar nach dem Schreiben zu lesen, da die Redis-Replikation asynchron ist. Der WAIT-Befehl kann genutzt werden, um die Datensicherheit in der Praxis zu verbessern und Replikate dazu zu zwingen, Schreibvorgänge zu bestätigen, bevor sie auf Clients antworten, was die Gesamtleistung beeinträchtigt. Die Verwendung von Replikatknoten für Lesevorgänge kann in Ihrer Clientbibliothek von ElastiCache für Redis konfiguriert werden, indem der ElastiCache-Reader-Endpunkt im deaktivierten Clustermodus verwendet wird. Wenn der Clustermodus aktiviert ist, verwenden Sie den Befehl READONLY von ElastiCache für Redis. Für viele Clientbibliotheken von ElastiCache

für Redis ist der Befehl READONLY von ElastiCache für Redis standardmäßig oder über eine Konfigurationseinstellung implementiert.

[Ressourcen]:

- [Verbindungsendpunkte ermitteln](#)
- [READONLY](#)
- [Erforderlich] Verwenden Sie Verbindungspooling. Der Aufbau einer TCP-Verbindung kostet sowohl auf Client- als auch auf Serverseite CPU-Zeit. Durch Pooling können Sie die TCP-Verbindung wiederverwenden.

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 Verbindungspooling über Ihre Clientbibliothek von ElastiCache für Redis (falls unterstützt) mit einem Framework implementieren, das 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. Dadurch werden Latenz und CPU-Auslastung reduziert, die erforderlich sind, um Verbindungen zu öffnen und zu schließen, und es wird ein TLS-Handshake durchgeführt, wenn TLS im Cluster aktiviert ist.

[Ressourcen]: [Konfigurieren von Amazon ElastiCache für Redis für eine höhere Verfügbarkeit](#)

- [Gut] Der Einsatz von Pipelining (sofern Ihre Anwendungsfälle dies zulassen) kann die Leistung erheblich steigern.
 - Mit Pipelining reduzieren Sie die Round-Trip Time (RTT, Roundtrip-Zeit) zwischen Ihren Anwendungsclients und dem Cluster und 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 stellen Sie ElastiCache-Komponenten für einen Workload bereit?

Einführung auf Fragenebene: ElastiCache-Umgebungen können manuell über die AWS-Konsole oder programmgesteuert über APIs, CLI, Toolkits usw. bereitgestellt werden. Bewährte Methoden für Operational Excellence sehen vor, Bereitstellungen möglichst mithilfe von Code zu automatisieren. Darüber hinaus können ElastiCache-Cluster entweder nach Workload isoliert oder zu Kostenoptimierungszwecken kombiniert werden.

Vorteil auf Fragenebene: Die Wahl des am besten geeigneten Bereitstellungsmechanismus für Ihre ElastiCache-Umgebungen kann die Operation Excellence im Laufe der Zeit verbessern. 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-Isolation verstehen, können Sie entweder dedizierte ElastiCache-Umgebungen pro Workload verwenden oder mehrere Workloads zu einzelnen Clustern kombinieren oder Kombinationen davon zu verwenden. Die Nachteile zu verstehen, kann dazu beitragen, ein Gleichgewicht zwischen Operational Excellence und Kostenoptimierung zu finden

- [Erforderlich] Machen Sie sich mit den für ElastiCache verfügbaren Bereitstellungsoptionen vertraut und automatisieren Sie diese Verfahren möglichst. Automatisierungsmöglichkeiten sind beispielsweise CloudFormation, AWS CLI/SDK und APIs.

[Ressourcen]:

- [Ressourcentypreferenz für Amazon ElastiCache](#)
- [elasticache](#)
- [Amazon-ElastiCache-API-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 eine detaillierte Kontrolle über Zugriff, Größe, Skalierung und Verwaltung von ElastiCache-Ressourcen pro Workload.

- [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 beinhaltet die Antizipation von Ausfällen durch regelmäßige „Pre-Mortem“-Übungen, um potenzielle Fehlerquellen zu identifizieren, damit diese beseitigt oder gemindert werden können. ElastiCache bietet eine Failover-API, die simulierte Knotenausfallereignisse zu Testzwecken 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 Dev/Test-Konten durch. [TestFailover](#)

OE 7: Wie beheben Sie Probleme mit Redis-Engine-Ereignissen?

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. Amazon ElastiCache für Redis kann Redis-Engine-Protokolle sowohl an Amazon CloudWatch als auch an Amazon Kinesis Data Firehose ausgeben.

Vorteil auf Fragenebene: Die Aktivierung von Redis-Engine-Protokollen in Clustern von Amazon ElastiCache für Redis bietet Einblicke in Ereignisse, die sich auf den Zustand und die Leistung von Clustern auswirken. Redis-Engine-Protokolle liefern Daten direkt aus der Redis-Engine, die über den ElastiCache-Ereignismechanismus nicht verfügbar sind. Durch die sorgfältige Beobachtung sowohl der ElastiCache-Ereignisse (siehe oben OE-1) als auch der Redis-Engine-Protokolle ist es möglich, eine Reihenfolge der Ereignisse bei der Fehlerbehebung sowohl aus Sicht des ElastiCache-Services als auch aus Sicht der Redis-Engine zu bestimmen.

- [Erforderlich] Stellen Sie sicher, dass die Redis-Engine-Protokollfunktion aktiviert ist. Diese ist ab ElastiCache für Redis 6.2 und höher 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 Redis-Engine-Protokolle ist.
- Wählen Sie in CloudWatch oder Kinesis Data Firehose ein geeignetes Zielprotokoll 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](#)
- Protokollziele: [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 Redis-Engine-Protokoll nach wichtigen Informationen abzufragen.

Erstellen Sie beispielsweise eine Abfrage für die CloudWatch-Protokollgruppe mit den Redis-Engine-Protokollen, die Ereignisse mit dem LogLevel 'WARNING' zurückgeben, wie zum Beispiel:

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Ressourcen]: [Analysieren von Protokolldaten mit Amazon CloudWatch Logs Insights](#)

Säule der Sicherheit 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

- [SEC 1: Welche Schritte unternehmen Sie, um den autorisierten Zugriff auf ElastiCache-Daten zu kontrollieren?](#)
- [SEC 2: Benötigen Ihre Anwendungen zusätzlich zu den netzwerkbasieren Kontrollen eine weitere Autorisierung für ElastiCache?](#)

- [SEC 3: Besteht das Risiko, dass Befehle versehentlich ausgeführt werden können und Datenverlust oder -ausfall verursachen?](#)
- [SEC 4: Wie stellen Sie mit ElastiCache sicher, dass gespeicherte Daten verschlüsselt werden?](#)
- [SEC 5: Wie verschlüsseln Sie Daten bei der Übertragung mit ElastiCache?](#)
- [SEC 6: Wie schränken Sie den Zugriff auf Ressourcen der Steuerebene ein?](#)
- [SEC 7: Wie erkennen Sie Sicherheitsereignisse und wie reagieren Sie darauf?](#)

SEC 1: Welche Schritte unternehmen Sie, um den autorisierten Zugriff auf ElastiCache-Daten zu kontrollieren?

Einführung auf Fragenebene: Alle ElastiCache-Cluster sind für den Zugriff von Instances von Amazon Elastic Compute Cloud in einer VPC, Serverless-Funktionen (AWS Lambda) oder Containern (Amazon Elastic Container Service) konzipiert. Das am häufigsten vorkommende Szenario ist der Zugriff auf einen ElastiCache-Cluster von einer Instance von Amazon Elastic Compute Cloud in derselben Amazon Virtual Private Cloud (Amazon VPC). Bevor Sie von einer Amazon-EC2-Instance eine Verbindung mit einem Cluster herstellen können, müssen Sie der Amazon-EC2-Instance Zugriff auf den Cluster gewähren. Um auf einen ElastiCache-Cluster zuzugreifen, der in einer VPC ausgeführt wird, muss dem Cluster Netzwerkeingang gewährt werden.

Vorteil auf Fragenebene: Der Netzwerkeingang für den Cluster wird über VPC-Sicherheitsgruppen gesteuert. Eine Sicherheitsgruppe dient als virtuelle Firewall für Ihre Amazon-EC2-Instances zur Steuerung von ein- und ausgehendem Datenverkehr. Eingehende Regeln steuern den eingehenden Datenverkehr zu Ihrer Instance und ausgehende Regeln steuern den ausgehenden Datenverkehr von Ihrer Instance. Im Fall von ElastiCache muss beim Starten eines Clusters 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 ist ElastiCache so konfiguriert, dass es ausschließlich in privaten Subnetzen bereitgestellt wird, sodass auf sie nur über das private Netzwerk der VPC 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 kein Eingangspfad zu ElastiCache. Um dies zu ändern, konfigurieren Sie für die Sicherheitsgruppe eine Regel für eingehenden Datenverkehr, welche die Quell-IP-Adresse/den Quell-IP-Bereich, den TCP-Verkehr und den Port für Ihren ElastiCache-Cluster angibt (z. B. Standardport 6379 für ElastiCache für Redis). Es ist zwar möglich, eine sehr breite Palette von Eingangsquellen zuzulassen, wie z. B. alle Ressourcen

innerhalb einer VPC (0.0.0.0/0). Es wird jedoch empfohlen, bei der Definition der eingehenden Regeln so detailliert wie möglich vorzugehen, z. B. nur eingehenden Zugriff für Redis-Clients zu autorisieren, die auf Amazon-EC2-Instances laufen, welche einer bestimmten Sicherheitsgruppe zugeordnet sind.

[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] Richtlinien von AWS Identity and Access Management können AWS Lambda-Funktionen zugewiesen werden, die ihnen den Zugriff auf ElastiCache-Daten ermöglichen. Um diese Funktion zu aktivieren, erstellen Sie eine IAM-Ausführungsrolle mit der entsprechenden `AWSLambdaVPCLambdaAccessExecutionRole`-Berechtigung und weisen Sie die Rolle dann der AWS Lambda-Funktion zu.

[Ressourcen]: Konfigurieren einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC: [Tutorial: Konfigurieren einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC](#)

SEC 2: Benötigen Ihre Anwendungen zusätzlich zu den netzwerkbasierenden Kontrollen eine weitere Autorisierung für ElastiCache?

Einführung auf Fragenebene: In Szenarien, in denen der Zugriff auf Cluster von ElastiCache für Redis auf individueller Client-Ebene eingeschränkt oder kontrolliert werden muss, wird empfohlen, sich mit dem Befehl `AUTH` von ElastiCache für Redis zu authentifizieren. Authentifizierungstoken von ElastiCache für Redis mit optionaler Benutzer- und Benutzergruppenverwaltung ermöglichen es ElastiCache für Redis, ein Passwort zu verlangen, bevor Clients Befehle und Zugriffsschlüssel ausführen können. Dadurch wird die Sicherheit auf der Datenebene verbessert.

Vorteil auf Fragenebene: ElastiCache für Redis bietet Mechanismen zum Schutz Ihrer Daten vor unbefugtem Zugriff. Dazu gehört auch, zu erzwingen, dass das Role-Based Access Control (RBAC)-Token `AUTH` oder `AUTH`-Token (Passwort) von Clients verwendet wird, um sich mit ElastiCache zu verbinden, bevor autorisierte Befehle ausgeführt werden.

- [Am besten] Legen Sie für ElastiCache für Redis 6.x und höher Authentifizierungs- und Autorisierungskontrollen fest, indem Sie Benutzergruppen, Benutzer und Zugriffszeichenfolgen

definieren. Weisen Sie Benutzer Benutzergruppen zu und weisen Sie Benutzergruppen dann Clustern zu. Damit RBAC verwendet wird, muss es bei der Clustererstellung ausgewählt und die Verschlüsselung während der Übertragung aktiviert sein. Stellen Sie sicher, dass Sie einen Redis-Client verwenden, der TLS unterstützt, um RBAC nutzen zu können.

[Ressourcen]:

- [Anwenden von RBAC auf eine Replikationsgruppe für ElastiCache für Redis](#)
 - [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#)
 - [ACL](#)
 - [Unterstützte Versionen von ElastiCache für Redis](#)
- [Am besten] Für frühere Versionen von ElastiCache für Redis als 6.x empfiehlt es sich, nicht nur ein starkes Token/Passwort festzulegen und eine strenge Passworrichtlinie für das AUTH-Token von ElastiCache für Redis einzuhalten, sondern auch das Passwort/Token zu wechseln. ElastiCache kann bis zu zwei (2) Authentifizierungstoken gleichzeitig verwalten. Sie können den Cluster auch so ändern, dass explizit die Verwendung von Authentifizierungstoken erforderlich ist.

[Resources]: [Ändern des AUTH-Tokens für einen vorhandenen Cluster von ElastiCache für Redis](#)

SEC 3: Besteht das Risiko, dass Befehle versehentlich ausgeführt werden können und Datenverlust oder -ausfall verursachen?

Einführung auf Fragenebene: Es gibt eine Reihe von Redis-Befehlen, die negative Auswirkungen auf den Betrieb haben 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 in einer Entwicklungsumgebung routinemäßig den Befehl FLUSHALL aufrufen und aufgrund eines Fehlers versehentlich versuchen, diesen Befehl in einem Produktionssystem aufzurufen, was zu unbeabsichtigtem Datenverlust führt.

Vorteil auf Fragenebene: Ab ElastiCache für Redis 5.0.3 in ElastiCache haben Sie die Möglichkeit, bestimmte Befehle umzubenennen, die sich störend auf Ihren Workload auswirken könnten. Durch das Umbenennen der Befehle kann verhindert werden, dass diese versehentlich auf dem Cluster ausgeführt werden.

- [Erforderlich]

[Ressourcen]:

- [ElastiCache für Redis Version 5.0.3 \(veraltet, Version 5.0.6 verwenden\)](#)

- [Parameteränderungen von Redis 5.0.3](#)
- [Redis-Sicherheit](#)

SEC 4: Wie stellen Sie mit ElastiCache sicher, dass gespeicherte Daten verschlüsselt werden?

Einführung auf Fragenebene: ElastiCache für Redis ist zwar ein In-Memory-Datenspeicher, es ist jedoch möglich, alle Daten zu verschlüsseln, die im Rahmen der Standardoperationen des Clusters (im Speicher) persistent gespeichert werden können. 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 für Redis bietet optionale Verschlüsselung im Ruhezustand, um die Datensicherheit zu erhöhen.

- [Erforderlich] Verschlüsselung im Ruhezustand kann nur für einen ElastiCache-Cluster (Replikationsgruppe) aktiviert werden, wenn dieser erstellt wird. Ein vorhandener Cluster kann nicht geändert werden, um mit der Verschlüsselung von Daten im Ruhezustand zu beginnen. Standardmäßig stellt ElastiCache die Schlüssel bereit, die für die Verschlüsselung im Ruhezustand verwendet werden, und verwaltet sie.

[Ressourcen]:

- [Bedingungen für die Verschlüsselung im Ruhezustand](#)
- [Aktivieren der Verschlüsselung im Ruhezustand](#)
- [Am besten] Nutzen Sie Instance-Typen von Amazon EC2, die Daten verschlüsseln, während sie sich im Speicher befinden (wie M6g oder R6g). Wenn möglich, sollten Sie erwägen, eigene Schlüssel für die Verschlüsselung im Ruhezustand zu verwalten. Für strengere Datensicherheitsumgebungen kann AWS Key Management Service (KMS) zur Selbstverwaltung von Customer Master Keys (CMK, Kundenmasterschlüssel) verwendet werden. Durch die ElastiCache-Integration in AWS Key Management Service können Sie die Schlüssel, die für die Verschlüsselung von Daten im Ruhezustand für Ihren Cluster von ElastiCache für Redis verwendet werden, erstellen, besitzen und verwalten.

[Ressourcen]:

- [Verwendung von kundenverwalteten Schlüsseln aus AWS Key Management Service](#)

- [AWS Key Management Service](#)
- [AWS-KMS-Konzepte](#)

SEC 5: Wie verschlüsseln Sie Daten bei der Übertragung mit ElastiCache?

Einführung auf Fragenebene: Es ist eine gängige Anforderung, zu verhindern, dass Daten während der Übertragung beschädigt werden. Dies stellt Daten innerhalb von Komponenten eines verteilten Systems sowie zwischen Anwendungsclients und Clusterknoten dar. ElastiCache für Redis unterstützt diese Anforderung, indem es die Verschlüsselung von Daten während der Übertragung zwischen Clients und Clustern sowie zwischen den Clusterknoten selbst ermöglicht. Instance-Typen der M6g- und R6g-Familien verfügen außerdem über eine ständig aktive In-Memory-Verschlüsselung.

Vorteil auf Fragenebene: Die Verschlüsselung während der Übertragung mit Amazon ElastiCache ist eine optionale Funktion, mit der Sie die Sicherheit Ihrer Daten an den am stärksten gefährdeten Stellen erhöhen können – wenn sie von einem Ort zum anderen übertragen werden.

- [Erforderlich] Verschlüsselung bei der Übertragung kann nur für einen ElastiCache-Cluster (Replikationsgruppe) aktiviert werden, wenn dieser erstellt wird. 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 der Verschlüsselung bei der Übertragung zu vergleichen.

[Ressourcen]:

- [Übersicht über die Verschlüsselung bei der Übertragung](#)

SEC 6: Wie schränken Sie den Zugriff auf Ressourcen der Steuerebene ein?

Einführung auf Fragenebene: IAM-Richtlinien und ARN ermöglichen differenzierte Zugriffskontrollen für ElastiCache für Redis und somit eine strengere Kontrolle bei der Verwaltung der Erstellung, Änderung und Löschung von Clustern von ElastiCache für Redis.

Vorteil auf Fragenebene: Die Verwaltung von Amazon-ElastiCache-Ressourcen wie Replikationsgruppen, Knoten usw. kann auf AWS-Konten beschränkt werden, die auf der Grundlage von IAM-Richtlinien über bestimmte Berechtigungen verfügen. Dadurch werden Sicherheit und Zuverlässigkeit der Ressourcen verbessert.

- [Erforderlich] Verwalten Sie den Zugriff auf Amazon-ElastiCache-Ressourcen, indem Sie AWS-Benutzern spezifische AWS Identity and Access Management-Richtlinien zuweisen, sodass Sie besser kontrollieren können, welche Konten welche Aktionen in Clustern ausführen können.

[Ressourcen]:

- [Übersicht über die Verwaltung von Zugriffsberechtigungen für Ihre ElastiCache-Ressourcen](#)
- [Verwenden identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#)

SEC 7: Wie erkennen Sie Sicherheitsereignisse und wie reagieren Sie darauf?

Einführung auf Fragenebene: Wenn ElastiCache mit aktiviertem RBAC bereitgestellt wird, exportiert es CloudWatch-Metriken, um Benutzer über Sicherheitsereignisse zu informieren. Diese Metriken helfen bei der Identifizierung von fehlgeschlagenen Authentifizierungsversuchen, Zugriffsschlüsseln oder der Ausführung von Befehlen, für die RBAC-Benutzer, die eine Verbindung herstellen, nicht autorisiert sind.

Darüber hinaus tragen AWS-Produkt- und -Serviceressourcen dazu bei, Ihren gesamten Workload zu schützen, indem sie Bereitstellungen automatisieren und alle Aktionen und Änderungen zur späteren 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-Metriken vertraut, die sich auf RBAC-Authentifizierungs- und Autorisierungsfehler beziehen.
 - AuthenticationFailures = fehlgeschlagene Versuche, sich bei Redis zu authentifizieren
 - KeyAuthorizationFailures = fehlgeschlagene Versuche von Benutzern, ohne Erlaubnis auf Schlüssel zuzugreifen
 - CommandAuthorizationFailures = fehlgeschlagene Versuche von Benutzern, Befehle ohne Erlaubnis auszuführen

[Ressourcen]:

- [Metriken für Redis](#)
- [Am besten] Es wird empfohlen, Warnmeldungen und Benachrichtigungen für diese Metriken einzurichten und je nach Bedarf zu reagieren.

[Ressourcen]:

- [Verwenden von Amazon CloudWatch-Alarmen](#)
- [Am besten] Verwenden Sie den Redis-Befehl ACL LOG, um weitere Informationen zu erhalten.

[Ressourcen]:

- [ACL LOG](#)
- [Am besten] Machen Sie sich mit den Funktionen der AWS-Produkte und -Services vertraut, die sich auf die Überwachung, Protokollierung und Analyse von ElastiCache-Bereitstellungen und -Ereignissen beziehen.

[Ressourcen]:

- [Protokollieren von API-Aufrufen von Amazon ElastiCache mithilfe von AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-backup-check](#)
- [Überwachung der Nutzung mithilfe von CloudWatch-Metriken](#)

Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens

Themen

- [REL 1: Wie unterstützen Sie Bereitstellungen von Hochverfügbarkeitsarchitekturen \(HA\)?](#)
- [REL 2: Wie erreichen Sie Ihre Recovery Point Objectives \(RPOs\) mit ElastiCache?](#)
- [REL 3: Wie unterstützen Sie die Anforderungen an Disaster Recovery \(DR, Notfallwiederherstellung\)?](#)
- [REL 4: Wie planen Sie Failovers effektiv?](#)
- [REL 5: Sind Ihre ElastiCache-Komponenten auf Skalierung ausgelegt?](#)

REL 1: Wie unterstützen Sie Bereitstellungen von Hochverfügbarkeitsarchitekturen (HA)?

Einführung auf Fragenebene: Wenn Sie die Hochverfügbarkeitsarchitektur von Amazon ElastiCache verstehen, können Sie bei Verfügbarkeitsereignissen in einem resilienten Zustand arbeiten.

Vorteil auf Fragenebene: Wenn Sie Ihre ElastiCache-Cluster so resilient gestalten, dass sie ausfallsicher sind, wird eine höhere Verfügbarkeit Ihrer ElastiCache-Bereitstellungen gewährleistet.

- [Erforderlich] Ermitteln Sie den Zuverlässigkeitsgrad, den Sie für Ihren ElastiCache-Cluster benötigen. 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: Memcached im Vergleich zu ElastiCache für Redis

1. Memcached bietet keinen Replikationsmechanismus und wird hauptsächlich für kurzlebige Workloads verwendet.
 2. ElastiCache für Redis bietet HA-Funktionen, die im Folgenden beschrieben werden.
- [Am besten] Verwenden Sie ElastiCache für Redis für Workloads, die HA erfordern, im Clustermodus mit mindestens zwei Replikaten pro Shard, selbst für Workloads mit geringen Durchsatzanforderungen, 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 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).

Amazon ElastiCache für Redis hat die Leistung auf diesen Knoten optimiert. Dadurch erhalten Sie eine bessere Replikations- und Synchronisierungsleistung, was zu einer insgesamt verbesserten Verfügbarkeit führt.

5. Überwachen und passen Sie die Größe an, um mit erwarteten Datenverkehrsspitzen umzugehen: Bei hoher Auslastung reagiert die Engine von ElastiCache für Redis möglicherweise nicht mehr, was sich auf die Verfügbarkeit auswirkt. `BytesUsedForCache` und `DatabaseMemoryUsagePercentage` sind gute Indikatoren für Ihre Speicherauslastung, während `ReplicationLag` ein Indikator für den Zustand Ihrer Replikation ist, der auf Ihrer Schreibrate basiert. Sie können diese Metriken verwenden, um die Clusterskalierung auszulösen.

6. Stellen Sie die clientseitige Resilienz sicher, indem Sie [vor einem Produktions-Failover-Ereignis Tests mit der Failover-API](#) durchführen.

[Ressourcen]:

- [Konfigurieren von Amazon ElastiCache für Redis für eine höhere Verfügbarkeit](#)
- [Hohe Verfügbarkeit mit Replikationsgruppen](#)

REL 2: Wie erreichen Sie Ihre Recovery Point Objectives (RPOs) mit ElastiCache?

Einführung auf Fragenebene: Machen Sie sich mit Workload-RPO vertraut, um fundierte Entscheidungen über ElastiCache-Backup- und Wiederherstellungsstrategien treffen zu können.

Vorteil auf Fragenebene: Eine integrierte RPO-Strategie kann die Geschäftskontinuität im Falle eines Notfallwiederherstellungs-Szenarios verbessern. Die Gestaltung Ihrer Sicherungs- und Wiederherstellungsrichtlinien kann Ihnen helfen, Ihre Recovery Point Objectives (RPO) für Ihre ElastiCache-Daten zu erreichen. ElastiCache für Redis bietet Snapshot-Funktionen, die in Amazon S3 gespeichert sind, sowie eine konfigurierbare 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 dem RPO Ihrer ElastiCache-Bereitstellungen vertraut und dokumentieren Sie es.
 - Beachten Sie, dass Memcached keine Backup-Prozesse anbietet.
 - Informieren Sie sich über die Möglichkeiten 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 in einem definierten Zeitfenster zu sichern.
 - Testen Sie regelmäßig Wiederherstellungen aus Ihren Backups.
- [Ressourcen]:
 - [Redis](#)
 - [Backup und Wiederherstellung für ElastiCache für Redis](#)
 - [Erstellen manueller Backups](#)
 - [Planen automatischer Backups](#)
 - [Backup und Wiederherstellung von Clustern von ElastiCache für Redis](#)

REL 3: Wie unterstützen Sie die Anforderungen an Disaster Recovery (DR, Notfallwiederherstellung)?

Einführung auf Fragenebene: Die Notfallwiederherstellung ist ein wichtiger Aspekt jeder Workload-Planung. ElastiCache für Redis bietet verschiedene Optionen zur Implementierung einer Notfallwiederherstellung auf der Grundlage der Anforderungen an die Belastbarkeit von Workloads. Mit Amazon ElastiCache für Redis Global Datastore können Sie in Ihren Cluster von ElastiCache für Redis in einer Region schreiben und die Daten aus zwei anderen regionsübergreifenden Replikatclustern zur Verfügung stellen. Dadurch sind Lesevorgänge mit geringer Latenz und eine regionsübergreifende Notfallwiederherstellung möglich.

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 alle Ihre ElastiCache-Komponenten auf der Grundlage der Workload-Anforderungen. ElastiCache ist insofern einzigartig, als einige Anwendungsfälle nur 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 der Clustermodus in Multi-AZ-Architekturen aktiviert ist und mindestens 3 AZs verfügbar sind.
- 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](#)
 - [Replikation über AWS-Regionen hinweg mit globalen Datenspeichern](#)
 - [Wiederherstellen aus einem Backup mit optionaler Cluster-Größenanpassung](#)
 - [Minimieren von Ausfallzeiten in ElastiCache für Redis mit Multi-AZ](#)

REL 4: Wie planen Sie Failovers effektiv?

Einführung auf Fragenebene: Die Aktivierung von Multi-AZ mit automatischen Failover ist eine bewährte Methode für ElastiCache. In bestimmten Fällen ersetzt ElastiCache für Redis die Primärknoten im Rahmen von Serviceoperationen. Beispiele hierfür sind geplante Wartungsereignisse und der unwahrscheinliche Falle eines Ausfalls des Knotens oder eines Problems mit der Availability Zone. Erfolgreiche Failover hängen sowohl von ElastiCache als auch von Ihrer Clientbibliothekskonfiguration ab.

Vorteil auf Fragenebene: Wenn Sie die bewährten Methoden für ElastiCache-Failover in Verbindung mit Ihrer spezifischen Clientbibliothek von ElastiCache für Redis 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 Clientbibliothek von ElastiCache für Redis 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]:

- [Minimieren von Ausfallzeiten in ElastiCache für Redis mit Multi-AZ](#)
- Sehen Sie sich die bewährten Methoden Ihrer Clientbibliothek von ElastiCache für Redis an.
- [Erforderlich] Erfolgreiche Failover hängen von einer fehlerfreien Replikationsumgebung zwischen dem Primärknoten und den Replikatknoten ab. Informieren Sie sich über den asynchronen Charakter der Redis-Replikation und machen Sie sich mit den verfügbaren CloudWatch-Metriken vertraut, um über die Replikationsverzögerung zwischen Primär- und Replikatknoten zu berichten. Für Anwendungsfälle, die eine höhere Datensicherheit erfordern, können Sie den Redis-Befehl WAIT nutzen, um Replikate zu zwingen, Schreibvorgänge zu bestätigen, bevor sie auf verbundene Clients antworten.

[Ressourcen]:

- [Metriken für Redis](#)
- [Überwachung bewährter Methoden mit Amazon ElastiCache für Redis unter Verwendung von Amazon CloudWatch](#)
- [Am besten] Überprüfen Sie regelmäßig die Reaktionsfähigkeit Ihrer Anwendung während des Failovers mithilfe der Test-Failover-API von ElastiCache.

[Ressourcen]:

- [Testen des automatischen Failovers auf einer Read Replica in Amazon ElastiCache für Redis](#)
- [Testen des automatischen Failovers](#)

REL 5: Sind Ihre ElastiCache-Komponenten auf Skalierung ausgelegt?

Einführung auf Fragenebene: Wenn Sie die Skalierungsfunktionen und die verfügbaren Bereitstellungstopologien verstehen, können sich Ihre ElastiCache-Komponenten im Laufe der Zeit an wechselnde Workload-Anforderungen anpassen. ElastiCache bietet eine vierfache Skalierung: ab auf (horizontal) sowie hoch/herunter (vertikal).

Vorteil auf Fragenebene: Bewährte Methoden für ElastiCache-Bereitstellungen zu befolgen, bietet nicht nur ein Höchstmaß an Skalierungsflexibilität, sondern erfüllt auch 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.
 - Für mehr READIOPS: Replikate hinzufügen
 - Für mehr WRITEOPS: Shards hinzufügen (aufskalieren)
 - Für mehr Netzwerk-I/O – netzwerkoptimierte Instances verwenden, hochskalieren
- [Am besten] Stellen Sie Ihre ElastiCache-Komponenten mit aktiviertem Clustermodus bereit, wobei mehr kleinere Knoten gegenüber weniger größeren Knoten bevorzugt werden. 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] Wenn der Clustermodus deaktiviert ist, nutzen Sie Lesereplikate, um die Gesamtlesekapazität zu erhöhen. ElastiCache unterstützt bis zu 5 Lesereplikate im deaktivierten Clustermodus sowie vertikale Skalierung.
- [Ressourcen]:
 - [Skalieren von Clustern von ElastiCache für Redis](#)
 - [Online-Hochskalieren](#)
 - [Skalieren von Clustern von ElastiCache für Memcached](#)

Amazon ElastiCache – Well-Architected Lens – Säule der Leistungseffizienz

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 Amazon-ElastiCache-Clusters?](#)
- [PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache-Cluster-Knoten?](#)
- [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 ausgeführte Befehle in Ihrem Amazon-ElastiCache-Cluster?](#)
- [PE 8: Inwieweit trägt Auto Scaling dazu bei, die Leistung des ElastiCache-Clusters zu steigern?](#)

PE 1: Wie überwachen Sie die Leistung Ihres Amazon-ElastiCache-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 ausführen, um sich ein Bild von den verfügbaren Metriken zu machen und eine Basisleistung festzulegen.
- [Am besten] Benennen Sie für Workloads von ElastiCache für Redis rechenintensive Befehle um, z. B. KEYS, damit Benutzer weniger Möglichkeiten haben, blockierende Befehle in Produktions-Clustern auszuführen.
 - Workloads von ElastiCache für Redis, auf denen Engine 6.x ausgeführt wird, kann die rollenbasierte Zugriffskontrolle nutzen, um bestimmte Befehle einzuschränken. Der Zugriff auf die Befehle kann gesteuert werden, indem Benutzer und Benutzergruppen mit der AWS-Konsole oder der CLI erstellt und die Benutzergruppen einem Cluster von ElastiCache für Redis zugeordnet werden. In Redis 6 können wir, wenn RBAC aktiviert ist, „- @dangerous“ verwenden. Dadurch werden teure Befehle wie KEYS, MONITOR, SORT usw. für diesen Benutzer unterbunden.

- Benennen Sie Befehle für Engine-Version 5.x mithilfe des `rename-commands`-Parameters in der Cluster-Parametergruppe von Amazon ElastiCache für Redis um.
- [Besser] Analysieren Sie langsame Abfragen und suchen Sie nach Optimierungsmöglichkeiten.
 - Im Zusammenhang mit Workloads von ElastiCache für Redis können Sie mehr über Ihre Abfragen erfahren, indem Sie das Slow-Protokoll analysieren. Sie können beispielsweise den Befehl `redis-cli slowlog get 10` verwenden, um die letzten 10 Befehle anzuzeigen, welche die Latenzschwellenwerte überschritten haben (standardmäßig 10 Sekunden).
 - Bestimmte Abfragen können mithilfe komplexer Datenstrukturen von ElastiCache für Redis 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.
 - Für Workloads von ElastiCache für Redis bietet `redis-benchmark` 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 mithilfe von CloudWatch-Metriken](#)
 - [Überwachung der Nutzung mithilfe von CloudWatch-Metriken](#)
 - [Verwenden von Amazon CloudWatch-Alarmen](#)
 - [Redis-spezifische Parameter](#)
 - [SLOWLOG](#)
 - [Redis-Benchmark](#)

PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache-Cluster-Knoten?

Einführung auf Fragenebene: Die Art und Weise, wie Ihre Anwendung eine Verbindung zu Amazon-ElastiCache-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] Lassen Sie die Clients eine Verbindung zum richtigen ElastiCache-Endpunkt herstellen.
 - Amazon ElastiCache für Redis implementiert je nach verwendetem Clustermodus unterschiedliche Endpunkte. Wenn der Clustermodus aktiviert ist, stellt ElastiCache einen Konfigurationsendpunkt bereit. Ist der Clustermodus deaktiviert, bietet ElastiCache einen primären Endpunkt, der normalerweise für Schreibvorgänge verwendet wird, und einen Reader-Endpunkt für die Verteilung der Lesevorgänge auf die Replikate. 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.
 - Für Memcached-Cluster mit mehreren Knoten bietet ElastiCache einen Konfigurationsendpunkt, der Auto Discovery ermöglicht. 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 von ElastiCache für Redis, um die Skalierbarkeit zu verbessern.
 - Cluster von ElastiCache für Redis (mit aktiviertem Clustermodus) unterstützen [Online-Skalierungsoperationen](#) (auf/ab und hoch/herunter), um Daten dynamisch auf 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 umverteilen, indem Sie Hashslots zwischen verfügbaren Shards in Ihrem Cluster von ElastiCache für Redis (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.
 - Berücksichtigen Sie die Auswirkungen multidimensionaler Redis-Datenstrukturen wie Listen, Streams, Sätze usw. Diese Datenstrukturen werden in einzelnen Redis-Schlüsseln gespeichert, die sich auf einem einzelnen 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. Für Workloads von ElastiCache für Redis können Sie Tastaturbefehle mithilfe von `redis-cli --hotkeys` erkennen, wenn eine LFU-Richtlinie für maximalen Arbeitsspeicher vorhanden ist.
- Erwägen Sie, Tastaturbefehle auf mehreren Knoten zu replizieren, um den Zugriff auf sie gleichmäßiger zu verteilen. Dieser Ansatz erfordert, dass der Client auf mehrere Primärknoten schreibt (der Redis-Knoten selbst bietet diese Funktionalität nicht) und zusätzlich zum ursprünglichen Schlüsselnamen eine Liste von Schlüsselnamen zum Lesen bereitstellt.
- ElastiCache für Redis Version 6 unterstützt servergestütztes [clientseitiges Caching](#). Dadurch können Anwendungen auf Änderungen an einem Schlüssel warten, bevor sie Netzwerkaufrufe an ElastiCache zurücksenden.
- [Ressourcen]:
 - [Konfigurieren von Amazon ElastiCache für Redis für eine höhere Verfügbarkeit](#)
 - [Verbindungsendpunkte ermitteln](#)
 - [Bewährte Methoden für den Lastausgleich](#)
 - [Online-Resharding und Shard-Rebalancing für Redis \(Clustermodus aktiviert\)](#)
 - [Clientseitiges Caching in Redis](#)

PE 3: Wie verfolgen und protokollieren Sie bei Caching-Workloads die Effektivität und Leistung Ihres Caches?

Einführung auf Fragenebene: Caching ist ein häufig anzutreffender Workload in 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 Redis](#)
- [Erforderlich] Messen und erfassen Sie die Leistung Ihres Anwendungscaches in Verbindung mit den Latenz- und CPU-Auslastungswerten, um zu ermitteln, ob Sie Anpassungen an Ihrem Time-to-Live-Wert oder an anderen Anwendungskomponenten vornehmen müssen. ElastiCache bietet eine Reihe von CloudWatch-Metriken für aggregierte Latenzen für jede Datenstruktur. Diese Latenzmetriken werden mithilfe der Commandstats-Statistik aus dem INFO-Befehl von ElastiCache für Redis berechnet und beinhalten weder das Netzwerk noch die E/A-Zeit. Dies ist lediglich die Zeit, die ElastiCache für Redis für die Verarbeitung der Operationen benötigt.

[Ressourcen]:

- [Metriken für Redis](#)
- [Überwachung bewährter Methoden mit Amazon ElastiCache für Redis 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](#)
- [Bewährte Methoden für das Caching](#)
- [Whitepaper „Skalierbare Leistung mit Amazon ElastiCache“](#)

PE 4: Wie optimiert Ihr Workload den Einsatz von Netzwerkressourcen und Verbindungen?

Einführung auf Fragenebene: ElastiCache für Redis und 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 mit Ihrem ElastiCache-Cluster.
 - 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 `currConnections` 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]:
 - [Amazon ElastiCache für Redis – globaler Datenspeicher](#)
 - [Memcached-spezifische Parameter](#)
 - [Amazon ElastiCache für Redis 5.0.3 verbessert das E/A-Handling, um die Leistung zu steigern](#)

- [Metriken für Redis](#)
- [Konfigurieren von Amazon ElastiCache für Redis für eine höhere Verfügbarkeit](#)

PE 5: Wie verwalten Sie die Löschung und/oder Bereinigung von Schlüsseln?

Einführung auf Fragenebene: Workloads haben unterschiedliche Anforderungen und ein anderes erwartetes Verhalten, wenn sich ein Clusterknoten den Grenzwerten für den Speicherverbrauch nähert. Amazon ElastiCache für Redis hat unterschiedliche Richtlinien für den Umgang mit diesen Situationen.

Vorteil auf Fragenebene: Die richtige Verwaltung des verfügbaren Speichers und das Verständnis der Löschrichtlinien 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 von `maxmemory-policy` in der Cluster-Parametergruppe von ElastiCache für Redis verwaltet werden.
 - Die Standardrichtlinie `volatile-lru` gibt Speicherplatz frei, indem Schlüssel mit einer festgelegten Ablaufzeit (TTL-Wert) bereinigt werden. Die Richtlinien „Least frequently used“ (LFU, am wenigsten häufig verwendet) und „Least recently used“ (LRU, am wenigsten zuletzt verwendet) entfernen Schlüssel je nach Nutzung.
 - Für Memcached-Workloads gibt es eine standardmäßige LRU-Richtlinie, welche die Bereinigungen auf jedem Knoten steuert. Die Anzahl der Bereinigungen in Ihrem Amazon-ElastiCache-Cluster kann mithilfe der Bereinigungsmetrik in Amazon CloudWatch überwacht werden.
- [Besser] Standardisieren Sie das Löschverhalten, um die Auswirkungen auf die Leistung Ihres Clusters zu kontrollieren und unerwartete Leistungsengpässe zu vermeiden.
 - Für Workloads von ElastiCache für Redis funktioniert UNLINK beim expliziten Entfernen von Schlüsseln aus dem Cluster wie DEL: Die angegebenen Schlüssel werden entfernt. 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 Workloads von ElastiCache für Redis 6.x kann das Verhalten des DEL-Befehls in der Parametergruppe mithilfe von `lazyfree-lazy-user-del`-Parametern geändert werden.
- [Ressourcen]:
 - [Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen](#)
 - [UNLINK](#)
 - [Cloud Financial Management mit AWS](#)

PE 6: Wie modellieren und interagieren Sie mit Daten in ElastiCache?

Einführung auf Fragenebene: ElastiCache ist stark anwendungsabhängig von den Datenstrukturen und dem verwendeten Datenmodell, muss aber auch den zugrundeliegenden Datenspeicher berücksichtigen (falls vorhanden). Machen Sie sich mit den verfügbaren Datenstrukturen von ElastiCache für Redis vertraut und stellen Sie sicher, dass Sie die für Ihre Anforderungen am besten geeigneten Datenstrukturen verwenden.

Vorteil auf Fragenebene: Die Datenmodellierung in ElastiCache umfasst mehrere Ebenen, einschließlich Anwendungsszenarien, Datentypen und Beziehungen zwischen Datenelementen. Darüber hinaus verfügt jeder Datentyp von ElastiCache für Redis und jeder Befehl über eigene, gut dokumentierte 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](#)
- [Am besten] Befehle von ElastiCache für Redis haben eine Zeitkomplexität, die durch die 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 für Redis führt eine große

Menge von Vorgängen mit hoher Zeitkomplexität zu einer geringeren Leistung und potenziellen Betriebszeitüberschreitungen.

[Ressourcen]:

- [Befehle](#)
- [Am besten] Verwenden Sie APIs, um grafische Einblicke in das Datenmodell in Ihrem Cluster zu erhalten.

[Ressourcen]:

- [Redis Commander](#)
- [Redis-Browser](#)
- [Redsmin](#)

PE 7: Wie protokollieren Sie langsam ausgeführte Befehle in Ihrem Amazon-ElastiCache-Cluster?

Einführung auf Fragenebene: Die Leistungsoptimierung bietet Vorteile durch die Erfassung, Aggregation und Benachrichtigung von Befehlen mit langer Laufzeit. Wenn Sie wissen, wie lange es dauert, bis Befehle ausgeführt werden, können Sie feststellen, welche Befehle zu einer schlechten Leistung führen und welche Befehle die Engine daran hindern, optimal zu arbeiten. Amazon ElastiCache für Redis bietet auch die Möglichkeit, diese Informationen an Amazon CloudWatch oder Amazon Kinesis Data Firehose weiterzuleiten.

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 für Redis mit Engine-Version 6.0 oder neuer, ordnungsgemäß konfigurierter Parametergruppe und aktivierter SLOWLOG-Protokollierung im Cluster.
 - Die erforderlichen Parameter sind nur verfügbar, wenn die Engine-Versionskompatibilität auf Redis Version 6.0 oder höher eingestellt ist.
 - Die SLOWLOG-Protokollierung erfolgt, wenn die Serverausführungszeit eines Befehls länger als ein bestimmter Wert dauert. 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 die Funktionen von CloudWatch oder Kinesis Data Firehose.

- Verwenden Sie die Filter- und Alarmfunktionen von CloudWatch, CloudWatch Logs Insights und Amazon Simple Notification Services, um die Leistungsüberwachung und die Benachrichtigung über Ereignisse zu gewährleisten.
- Verwenden Sie die Streaming-Funktionen von Kinesis Data Firehose, um SLOWLOG-Protokolle im permanenten Speicher zu archivieren oder um eine automatische Cluster-Parameteroptimierung auszulösen.
- Stellen Sie fest, ob das JSON- oder Plain-TEXT-Format Ihren Anforderungen am besten entspricht.
- Geben Sie IAM-Berechtigungen für die Veröffentlichung in CloudWatch oder Kinesis Data Firehose an.
- [Besser] Konfigurieren Sie `slowlog-log-slower-than` auf einen anderen Wert als den Standardwert.
 - Dieser Parameter bestimmt, wie lange ein Befehl innerhalb der Redis-Engine ausgeführt werden kann, bevor er als Befehl mit langer Laufzeit 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 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 Redis-CLI aus über den Befehl SLOWLOG analysiert werden müssen, sollten Sie erwägen, diesen Wert zu erhöhen. Dadurch können mehr Befehle im Redis-Speicher verbleiben.

Wenn Sie die SLOWLOG-Daten entweder an CloudWatch Logs oder Kinesis Data Firehose senden, bleiben die Daten erhalten und können außerhalb des ElastiCache-Systems analysiert werden. Dadurch wird die Notwendigkeit reduziert, eine große Anzahl langsam laufender Befehle im Redis-Speicher abzulegen.

- [Ressourcen]:

- [Wie aktiviere ich Redis Slow Log in einem Cache-Cluster von ElastiCache für Redis?](#)
- [Protokollzustellung](#)
- [Redis-spezifische Parameter](#)
- <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
- [Amazon Kinesis Data Firehose](#)

PE 8: Inwieweit trägt Auto Scaling dazu bei, die Leistung des ElastiCache-Clusters zu steigern?

Einführung auf Fragenebene: Durch die Implementierung der Funktion von Redis Auto Scaling können sich Ihre ElastiCache-Komponenten im Laufe der Zeit anpassen, um die Anzahl der gewünschten Shards oder Replikate 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 Spitzen des Workloads verstehen und entsprechend planen, können Sie die Caching-Leistung und die Geschäftskontinuität verbessern. Auto Scaling von ElastiCache für Redis überwacht kontinuierlich Ihre CPU-/Speicherauslastung, um sicherzustellen, dass Ihr Cluster mit den gewünschten Leistungsniveaus arbeitet.

- [Erforderlich] Beim Starten eines Clusters für ElastiCache für Redis:
 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]:

- [Skalieren von Clustern in Redis \(Clustermodus aktiviert\)](#)
- [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 entsprechende CloudWatch-Metriken für die Clusterauslastung über einen Zeitraum von vier Wochen analysieren, um den Schwellenwert für den Zielwert 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 mithilfe von CloudWatch-Metriken](#)
- [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](#)
- [Registrieren eines skalierbaren Ziels](#)

Säule der Kostenoptimierung von Amazon ElastiCache 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

- [COST 1: Wie ermitteln 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?](#)

- [COST 2: Wie verwenden Sie Tools zur kontinuierlichen Überwachung, um die mit Ihren ElastiCache-Ressourcen verbundenen Kosten zu optimieren?](#)
- [COST 3: Sollten Sie einen Instance-Typ verwenden, der Daten-Tiering unterstützt? Was sind die Vorteile von Daten-Tiering? Wann sollten Daten-Tiering-Instances nicht verwendet werden?](#)

COST 1: Wie ermitteln 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 von der Entwicklung über die Produktion bis hin zur Außerbetriebnahme erstellten Ressourcen verfolgen können, lassen sich die mit ElastiCache verbundenen Kosten besser kontrollieren.

Vorteil auf Fragenebene: Die kontinuierliche Erfassung aller mit Ihrem Workload verbundenen Kosten erfordert fundiertes Wissen über die Architektur, zu der ElastiCache als 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 Center of Excellence (CCoE, Cloud-Kompetenzzentrum) mit einer seiner Gründungschartas ein, das die Definition, Nachverfolgung und Umsetzung von Metriken rund um die ElastiCache-Nutzung Ihres Unternehmens übernimmt. Wenn ein CCoE vorhanden ist und funktioniert, stellen Sie sicher, dass es weiß, wie die mit ElastiCache verbundenen Kosten interpretiert und verfolgt werden können. Wenn Ressourcen erstellt werden, überprüfen Sie anhand von IAM-Rollen und -Richtlinien, dass 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. Das CCoE sollte Kostenkennzahlen identifizieren, definieren und veröffentlichen, die regelmäßig (monatlich) aktualisiert werden und sich auf die wichtigsten ElastiCache-Nutzungen beziehen, und zwar für kategoriale Daten wie:

- 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. Das CCoE besteht aus einem funktionsübergreifenden Team, in dem nicht ausschließlich Softwareentwicklungs-, Datenmanagement-, Produktteam-, Finanz- und Führungsteams in Ihrem Unternehmen vertreten sind.

[Ressourcen]:

- [Schaffen eines Cloud-Kompetenzzentrums](#)
 - [Amazon-ElastiCache-Preise](#)
- [Erforderlich] Verwenden Sie Kostenzuordnungs-Tags, um die Kosten auf einer niedrigen Granularitätsstufe zu verfolgen. Nutzen Sie AWS-Kostenmanagement, um Ihre AWS-Kosten und -Nutzung im Zeitverlauf zu visualisieren, zu verstehen und zu verwalten.
1. Verwenden Sie Tags, um Ihre Ressourcen zu organisieren, und Kostenzuordnungs-Tags, um Ihre AWS Kosten detailliert nachzuverfolgen. Nach dem Aktivieren der Kostenzuordnungs-Tags verwendet AWS diese zur Verwaltung Ihrer Ressourcenkosten im Bericht zur Kostenzuordnung und erleichtert es Ihnen so, Ihre AWS-Kosten zu kategorisieren und zu verfolgen. AWS bietet zwei Arten von Kostenzuordnungs-Tags, von AWS generierte Tags und benutzerdefinierte Tags. AWS definiert, erstellt und wendet die von AWS generierten Tags für Sie an und Sie definieren und erstellen benutzerdefinierte Tags und wenden diese an. Sie müssen beide Arten von Tags separat aktivieren, bevor sie in AWS-Kostenmanagement oder einem Kostenzuordnungsbericht angezeigt werden können.
 2. Verwenden Sie Kostenzuordnungs-Tags, um Ihre AWS-Rechnung so zu organisieren, dass sie Ihre eigene Kostenstruktur widerspiegelt. Wenn Sie Ihren Ressourcen in Amazon ElastiCache Kostenzuordnungs-Tags hinzufügen, können Sie Kosten verfolgen, indem Sie 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-Kostenzuordnungs-Tags](#)
- [Überwachen von Kosten mit Kostenzuordnungs-Tags](#)

- [AWS Cost Explorer](#)
- [Am besten] Setzen Sie die ElastiCache-Kosten mit Metriken in Beziehung, die das gesamte Unternehmen betreffen.
 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 mithilfe von CloudWatch-Metriken](#)
- [Gut] Sorgen Sie für einen aktuellen architektonischen und betrieblichen Überblick über Metriken und Kosten für den gesamten Workload, der ElastiCache verwendet.
 1. Machen Sie sich mit Ihrem gesamten Lösungsökosystem vertraut. ElastiCache ist in der Regel Teil eines vollständigen Ökosystems von AWS-Services in seiner Technologiepalette, von Clients bis hin zu API Gateway, Redshift und QuickSight für Berichtstools (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 Verwendung von Rechenleistung, Netzwerk, Speicher, Lebenszyklusrichtlinien, Erfassung von Metriken sowie die betrieblichen und funktionalen ElastiCache-Elemente Ihrer Anwendung enthalten.
 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 für eine effektive Kostenmanagementstrategie für Ihren ElastiCache von entscheidender Bedeutung.

COST 2: Wie verwenden 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 Anwendungsleistungsmetriken anstreben. Amazon CloudWatch bietet Einblicke in wichtige betriebliche Metriken, anhand derer Sie beurteilen können, ob Ihre ElastiCache-Ressourcen im Verhältnis zu Ihren Anforderungen über- oder unterausgelastet sind. Aus Sicht der Kostenoptimierung müssen Sie verstehen, wann Sie zu viel bereitstellen, und in der Lage sein, geeignete Mechanismen zu entwickeln, um die Größe Ihrer ElastiCache-Ressourcen zu ändern und gleichzeitig Ihre betrieblichen, Verfügbarkeits-, Resilienz- 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, dass sie über einen längeren Zeitraum betrieben werden.

- [Erforderlich] Verwenden Sie CloudWatch, um Ihre ElastiCache-Cluster zu überwachen und zu analysieren, wie sich diese Metriken auf Ihre Dashboards von AWS Cost Explorer beziehen.
 1. ElastiCache bietet sowohl Metriken auf Host-Ebene (z. B. CPU-Auslastung) als auch für die Cache-Engine-Software spezifische Metriken (z. B. Cache-Gets und Cache-Misses). Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.
 2. ElastiCache-Leistungsmetriken (CPUUtilization, EngineUtilization, SwapUsage, CurrConnections und Evictions) können darauf hinweisen, dass Sie hoch-/herunterskalieren (größere/kleinere Cache-Knotentypen verwenden) oder ab-/aufskalieren (mehr/weniger Shards hinzufügen) 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 CloudWatch-Metriken](#)
- [Welche Metriken sollte ich überwachen?](#)
- [Amazon-ElastiCache-Preise](#)
- [Erforderlich] Verstehen und dokumentieren Sie Ihre Backup-Strategie und die Auswirkungen auf die Kosten.

1. Mit ElastiCache werden die Backups in Amazon S3 gespeichert, was robusten Speicher 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 Anforderungen an reservierte Instances 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.

COST 3: Sollten Sie einen Instance-Typ verwenden, der Daten-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 wird, muss die Leerlaufzeit des ElastiCache-Objekts 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 des pro Instance verfügbaren SSD-Speichers unterstützen Daten-Tiering-Instances eine viel höhere Gesamtkapazität für Daten pro Instance.

- [Erforderlich] Kennen der Einschränkungen von Daten-Tiering-Instances
 1. Nur für Cluster von ElastiCache für Redis verfügbar.
 2. Nur begrenzte Instance-Typen unterstützen Daten-Tiering.
 3. Es wird nur ElastiCache für Redis Version 6.2 und höher unterstützt
 4. Große Objekte werden nicht auf SSD ausgelagert. Objekte über 128 MiB werden im Speicher aufbewahrt.

[Ressourcen]:

- [Daten-Tiering](#)
- [Amazon-ElastiCache-Preise](#)
- [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 ist mit geringen Leistungseinbußen verbunden, da diese auf eine lokale SSD 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 mithilfe von Nicht-Daten-Tiering-Instances betreiben, bis Sie Ihre Anforderungen besser verstehen.
 2. Analysieren Sie das Datennutzungsmuster Ihrer ElastiCache-Cluster.
 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](#)
- [Amazon-ElastiCache-Preise](#)

Häufige Schritte und bewährte Methoden zur Fehlerbehebung

Themen

- [Verbindungsprobleme](#)
- [Redis-Clientfehler](#)
- [Fehlerbehebung bei hoher Latenz in ElastiCache Serverless](#)
- [Fehlerbehebung bei Drosselungsproblemen in ElastiCache Serverless](#)
- [Verwandte Themen](#)

Verbindungsprobleme

Wenn Sie keine Verbindung zu Ihrem ElastiCache Cache herstellen können, sollten Sie einen der folgenden Punkte berücksichtigen:

1. Verwenden von TLS: Wenn Sie beim Versuch, eine Verbindung zu Ihrem ElastiCache Endpunkt herzustellen, eine Verbindung zum Aufhängen herstellen, verwenden Sie möglicherweise kein TLS in Ihrem Client. Wenn Sie ElastiCache Serverless verwenden, ist die Verschlüsselung während der Übertragung immer aktiviert. Stellen Sie sicher, dass Ihr Client TLS verwendet, um eine Verbindung zum Cache herzustellen. Weitere Informationen zum Herstellen einer Verbindung mit einem TLS-fähigen Cache finden Sie [unter](#) .
2. VPC: ElastiCache caches sind nur innerhalb einer VPC zugänglich. Stellen Sie sicher, dass die EC2-Instance, von der aus Sie auf den Cache zugreifen, und der ElastiCache Cache in derselben VPC erstellt werden. Alternativ müssen Sie das [VPC-Peering](#) zwischen der VPC, in der sich Ihre EC2-Instance befindet, und der VPC aktivieren, in der Sie Ihren Cache erstellen.
3. Sicherheitsgruppen: ElastiCache verwendet Sicherheitsgruppen, um den Zugriff auf Ihren Cache zu steuern. Berücksichtigen Sie dabei Folgendes:
 - a. Stellen Sie sicher, dass die von Ihrem ElastiCache Cache verwendete Sicherheitsgruppe eingehenden Zugriff auf sie von Ihrer EC2-Instance aus zulässt. Hier erfahren Sie <https://docs.aws.amazon.com/vpc/latest/userguide/security-group-rules.html>, 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 Cache ermöglicht (6379 und 6380 für Serverless und 6379 standardmäßig für selbst entwickelte Ports). ElastiCache verwendet diese Ports, um Redis-Befehle zu akzeptieren. Weitere Informationen zum Einrichten des Portzugriffs finden Sie [hier](#).

Redis-Clientfehler

ElastiCache Auf Serverless kann nur über Redis-Clients zugegriffen werden, die das Redis-Clustermodus-Protokoll unterstützen. Je nach Clusterkonfiguration können Sie von Redis-Clients in beiden Modi aus auf selbst entwickelte Cluster zugreifen.

Wenn in Ihrem Client Redis-Fehler auftreten, sollten Sie Folgendes berücksichtigen:

1. Cluster-Modus: Wenn CROSS Bol-Fehler oder -Fehler mit dem Befehl [SELECT](#) Redis auftreten, versuchen Sie möglicherweise, mit einem Redis-Client, der das Redis-Cluster-Protokoll nicht unterstützt, auf einen Cache mit aktiviertem Cluster-Modus zuzugreifen. ElastiCache Serverless unterstützt nur Redis-Clients, die das Redis-Cluster-Protokoll unterstützen. Wenn Sie Redis im „Clustermodus deaktiviert“ (CMD) verwenden möchten, müssen Sie Ihren eigenen Cluster entwerfen.

2. CROSS Slot-Fehler: Wenn der `ERR_CROSSSLOT Keys in request don't hash to the same slot` Fehler auftritt, versuchen Sie möglicherweise, auf Schlüssel zuzugreifen, die nicht zum selben Slot in einem Cluster-Modus-Cache gehören. Zur Erinnerung: ElastiCache Serverless arbeitet immer im Cluster-Modus. Multi-Schlüssel-Operationen, Transaktionen oder Lua-Skripte mit mehreren Schlüsseln sind nur zulässig, wenn sich alle beteiligten Schlüssel im selben Hash-Slot befinden.

Weitere bewährte Methoden zur Konfiguration von Redis-Clients finden Sie in diesem [Blogbeitrag](#).

Fehlerbehebung bei hoher Latenz in ElastiCache Serverless

Wenn Ihr Workload eine hohe Latenz aufweist, können Sie die CloudWatch `SuccessfulWriteRequestLatency` Metriken `SuccessfulReadRequestLatency` und analysieren, um zu überprüfen, ob die Latenz mit ElastiCache Serverless zusammenhängt. Diese Metriken messen die Latenz, die für ElastiCache Serverless intern ist – clientseitige Latenz und Netzwerklaufzeiten zwischen Ihrem Client und dem ElastiCache Serverless-Endpunkt sind nicht enthalten.

Einige Schwankungen und gelegentliche Spitzen sollten kein Grund zur Besorgnis sein. Wenn die Average Statistik jedoch einen starken Anstieg zeigt und anhält, sollten Sie die AWS Health Dashboard und Ihr Personal Health Dashboard überprüfen, um weitere Informationen zu erhalten. Erwägen Sie bei Bedarf, einen Support-Fall mit zu öffnen AWS Support.

Berücksichtigen Sie die folgenden bewährten Methoden und Strategien zur Reduzierung der Latenz:

- **Read from Replica aktivieren:** Wenn Ihre Anwendung dies zulässt, empfehlen wir, die Funktion „Read from Replica“ in Ihrem Redis-Client zu aktivieren, um Lesevorgänge zu skalieren und eine geringere Latenz zu erreichen. Wenn diese Option aktiviert ist, versucht ElastiCache Serverless, Ihre Leseanforderungen an Replikat-Cache-Knoten weiterzuleiten, die sich in derselben Availability Zone (AZ) wie Ihr Client befinden, wodurch eine AZ-übergreifende Netzwerklatenz vermieden wird. Beachten Sie, dass das Aktivieren der Read-from-Replica-Funktion in Ihrem Client bedeutet, dass Ihre Anwendung letztendliche Datenkonsistenz akzeptiert. Ihre Anwendung erhält möglicherweise einige Zeit ältere Daten, wenn Sie nach dem Schreiben in einen Schlüssel versuchen zu lesen.
- **Stellen Sie sicher, dass Ihre Anwendung in denselben AZs wie Ihr Cache bereitgestellt wird:** Sie können eine höhere clientseitige Latenz beobachten, wenn Ihre Anwendung nicht in denselben AZs wie Ihr Cache bereitgestellt wird. Wenn Sie einen Serverless-Cache erstellen, können Sie die Subnetze bereitstellen, von denen aus Ihre Anwendung auf den Cache zugreift, und ElastiCache Serverless erstellt VPC-Endpunkte in diesen Subnetzen. Stellen Sie sicher, dass Ihre Anwendung

in denselben AZs bereitgestellt wird. 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 Serverless-Anforderungen werden über eine TLS-fähige TCP-Verbindung unter Verwendung des RESP-Protokolls gestellt. Das Initiieren der Verbindung (einschließlich der Authentifizierung der Verbindung, falls konfiguriert) dauert Zeit, sodass die Latenz der ersten Anforderung höher ist als typisch. Anforderungen über eine bereits initialisierte Verbindung liefern die gleichbleibend niedrige Latenz ElastiCache von . Aus diesem Grund sollten Sie erwägen, Verbindungspooling zu verwenden oder vorhandene Redis-Verbindungen wiederzuverwenden.
- **Skalierungsgeschwindigkeit:** ElastiCache Serverless wird automatisch skaliert, wenn Ihre Anforderungsrate steigt. Ein plötzlicher starker Anstieg der Anforderungsrate, schneller als die Geschwindigkeit, mit der ElastiCache Serverless skaliert, kann für einige Zeit zu einer erhöhten Latenz führen. ElastiCache Serverless kann seine unterstützte Anforderungsrate in der Regel schnell erhöhen und bis zu 10–12 Minuten dauern, bis die Anforderungsrate verdoppelt wird.
- **Untersuchen von Befehlen mit langer Laufzeit:** Einige Redis-Befehle, einschließlich Lua-Skripts oder -Befehle für große Datenstrukturen, können lange laufen. Um diese Befehle zu identifizieren, ElastiCache veröffentlicht Metriken auf Befehlsebene. Mit [ElastiCache Serverless](#) können Sie die `-BasedECPUs` Metriken verwenden.
- **Gedrosselte Anforderungen:** Wenn Anforderungen in ElastiCache Serverless gedrosselt werden, kann es zu einer Erhöhung der clientseitigen Latenz in Ihrer Anwendung kommen. Wenn Anforderungen in ElastiCache Serverless gedrosselt werden, sollte die `ThrottledRequests` [ElastiCache Serverless](#)-Metrik steigen. Lesen Sie den folgenden Abschnitt zur Fehlerbehebung bei gedrosselten Anfragen.
- **Einheitliche Verteilung von Schlüsseln und Anforderungen:** In ElastiCache für Redis kann eine ungleichmäßige Verteilung von Schlüsseln oder Anforderungen pro Slot 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 Slot in einer Workload, die einfache SET/GET-Befehle ausführt. Wir empfehlen, Ihren Schlüssel und Ihre Anforderungsverteilung über Slots hinweg auszuwerten und eine einheitliche Verteilung zu gewährleisten, wenn Ihre Anforderungsrate dieses Limit überschreitet.

Fehlerbehebung bei Drosselungsproblemen in ElastiCache Serverless

In serviceorientierten Architekturen und verteilten Systemen wird die Begrenzung der Geschwindigkeit, mit der API-Aufrufe durch verschiedene Servicekomponenten verarbeitet werden, als Drosselung bezeichnet. Dadurch werden Spitzen geglättet, Abweichungen im Komponentendurchsatz gesteuert und vorhersehbarere Wiederherstellungen ermöglicht, wenn ein unerwartetes Betriebsereignis auftritt. ElastiCache Serverless ist für diese Architekturtypen konzipiert und die meisten Redis-Clients verfügen über integrierte 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 Anforderungen in ElastiCache Serverless gedrosselt werden, sollte die `ThrottledRequests` [ElastiCache Serverless](#)-Metrik steigen. Wenn Sie eine hohe Anzahl gedrosselter Anfragen feststellen, sollten Sie Folgendes berücksichtigen:

- **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 ElastiCache Serverless skaliert wird, können Ihre Anforderungen gedrosselt werden, während ElastiCache Serverless zur Anpassung an Ihren Workload skaliert. ElastiCache Serverless kann die Speichergröße in der Regel schnell erhöhen, was bis zu 10–12 Minuten dauert, bis die Speichergröße in Ihrem Cache verdoppelt wird.
- **Einheitliche Verteilung von Schlüsseln und Anforderungen:** In ElastiCache für Redis kann eine ungleichmäßige Verteilung von Schlüsseln oder Anforderungen pro Slot zu einem Hot Slot führen. Ein Hot-Slot kann zu einer Drosselung von Anforderungen führen, wenn die Anforderungsrate für einen einzelnen Slot in einem Workload, der einfache SET/GET-Befehle ausführt, 30.000 ECPUs/Sekunde überschreitet.
- **Aus Replikat lesen:** Wenn die Anwendung dies zulässt, sollten Sie die Funktion „Aus Replikat lesen“ verwenden. Die meisten Redis-Clients können so konfiguriert werden, dass Lesevorgänge „skaliert werden“, um Lesevorgänge an Replikatknoten weiterzuleiten. Mit dieser Funktion können Sie den Leseverkehr skalieren. Darüber hinaus leitet ElastiCache Serverless automatisch Lesevorgänge aus Replikatanforderungen an Knoten in derselben Availability Zone wie Ihre Anwendung weiter, was zu einer geringeren Latenz führt. Wenn Aus Replikat lesen aktiviert ist, können Sie bis zu 90.000 ECPUs/Sekunde auf einem einzigen Slot für Workloads mit einfachen SET/GET-Befehlen erreichen.

Verwandte Themen

- [Zusätzliche Schritte zur Fehlerbehebung](#)
- [the section called “Bewährte Methoden und Caching-Strategien”](#)

Zusätzliche Schritte zur Fehlerbehebung

Die folgenden Punkte müssen bei der Behebung anhaltender Verbindungsprobleme mit überprüft werden ElastiCache:

Themen

- [Sicherheitsgruppen](#)
- [Netzwerk-ACLs](#)
- [Routing-Tabellen](#)
- [DNS-Auflösung](#)
- [Identifizieren von Problemen mit serverseitiger Diagnose](#)
- [Netzwerkverbindbarkeit](#)
- [Netzwerkbezogene Grenzen](#)
- [CPU-Verwendung](#)
- [Verbindungen, die von der Serverseite beendet werden](#)
- [Clientseitige Fehlerbehebung für Amazon EC2 Instances](#)
- [Aufschlüsselung der Zeit, die zum Abschließen einer einzelnen Anfrage benötigt wird](#)

Sicherheitsgruppen

Sicherheitsgruppen sind virtuelle Firewalls, die Ihren ElastiCache Client (EC2-Instance, AWS Lambda Funktion, Amazon ECS-Container usw.) und ElastiCache Ihren 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 ElastiCache auf der 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 ausgehenden TCP-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 Redis. 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 einer Amazon VPC](#).

Netzwerk-ACLs

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. Netzwerk-ACLs werden Subnetzen zugewiesen, nicht bestimmten Ressourcen. Es ist möglich, der Client-Ressource dieselbe ACL ElastiCache zuzuweisen, insbesondere wenn sie sich im selben Subnetz befinden.

Standardmäßig erlauben Netzwerk-ACLs alle Traffic. Es ist jedoch möglich, sie anzupassen, um Datenverkehr zu verweigern oder zu erlauben. Darüber hinaus erfolgt die Auswertung der ACL-Regeln sequenziell, was bedeutet, dass die Regel mit der niedrigsten Zahl, die dem Datenverkehr entspricht, dies zulässt oder verweigert. Die Mindestkonfiguration für den Redis-Datenverkehr ist:

Clientseitige Netzwerk-ACL:

- Regeln für eingehenden Datenverkehr:
- Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
- Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
- 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;
- Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
- Protokoll: TCP
- Portbereich: 6379
- Quelle: 0.0.0.0/0 (oder die Cluster-Subnetze. ElastiCache Beachten Sie, dass die Verwendung bestimmter IPs im Falle eines Failovers oder der Skalierung des Clusters zu Problemen führen kann.
- Erlauben/Verweigern

ElastiCache Netzwerk-ACL:

- Regeln für eingehenden Datenverkehr:
 - Regelnummer: vorzugsweise niedriger als jede Ablehnungsregel;
 - Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
 - 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;
 - Type (Typ): Custom TCP Rule (TCP-Regel anpassen);
 - Protokoll: TCP
 - Portbereich: 1024-65535
 - Quelle: 0.0.0.0/0 (oder die Cluster-Subnetze. ElastiCache Beachten Sie, dass die Verwendung bestimmter IPs im Falle eines Failovers oder der Skalierung des Clusters zu Problemen führen kann.
 - Erlauben/Verweigern

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

Routing-Tabellen

Ähnlich wie Netzwerk-ACLs kann jedes Subnetz unterschiedliche Routentabellen 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.

Komplexere Umgebungen, die mehrere VPCs, dynamisches Routing oder Netzwerk-Firewalls umfassen, können schwierig zu beheben sein. Siehe [Netzwerkverbindbarkeit](#), um zu bestätigen, dass Ihre Netzwerkeinstellungen angemessen sind.

DNS-Auflösung

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 von der ElastiCache Engine sind gängige Informationsquellen zur Identifizierung potenzieller Ursachen von Verbindungsproblemen. Eine gute Analyse beginnt üblicherweise mit den folgenden Elementen:

- **CPU-Auslastung:** Redis ist eine Multi-Thread-Anwendung. Die Ausführung jedes Befehls erfolgt jedoch in einem einzigen (Haupt-) Thread. Aus diesem Grund stellt ElastiCache die Metriken `CPUtilization` und `EngineCPUtilization` bereit. `CPUtilization` stellt die CPU-Auslastung für den Redis-Prozess und `EngineCPUtilization` die Nutzung für alle vCPUs bereit. Knoten mit mehr als einer vCPU haben normalerweise unterschiedliche Werte für `CPUtilization` und `EngineCPUtilization`, die zweite ist allgemein höher. Hoch `EngineCPUtilization` kann durch eine erhöhte Anzahl von Anforderungen oder komplexe Vorgänge verursacht werden, die eine erhebliche CPU-Zeit in Anspruch nehmen. Sie können beide folgendermaßen identifizieren:
 - **Erhöhte Anzahl von Anforderungen:** Prüfen Sie auf Erhöhungen für andere Metriken, die den `EngineCPUtilization`-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 `EngineCPUtilization` 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 kann die Verwendung von Read-Replicas erfolgen, 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, die Replikatknoten erkennen und die Lesevorgänge über die [READONLY](#) Redis-Befehl, und senden Sie die Leseanforderungen an die Replikate.
 - **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 Behandeln von Verbindungen bedeutet einen erheblichen CPU-Overhead. Darüber hinaus wirkt sich der TCP-3-Wege-Handshake, der zum Erstellen neuer Verbindungen erforderlich ist, negativ auf die Gesamtreaktionszeiten 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` keine großen Variationen, und die `NewConnections` sollte so niedrig wie möglich sein. Redis bietet optimale Leistung mit einer geringen Anzahl von `CurrConnections`. Wenn Sie `CurrConnection` in der Größenordnung von zehn oder Hunderten halten, wird die Nutzung von Ressourcen minimiert, um einzelne Verbindungen wie Clientpuffer und CPU-Zyklen zu unterstützen, um die Verbindung zu bedienen.

- **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. Beispiele dafür sind Anwendungen mit einer hohen Rate kleiner Anforderungen eher die CPU-Auslastung als der Netzwerkdurchsatz, während größere Schlüssel eine höhere Netzwerkauslastung verursachen. 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 Redis mit aktiviertem Cluster-Modus, sodass die Schreibvorgänge auf mehrere Shards verteilt oder auf einen Knotentyp mit mehr Netzwerkfähigkeiten skaliert werden können.

- Die CloudWatchmetrics NetworkBytesIn und NetworkBytesOut geben jeweils die Datenmenge an, die in den Knoten eingeht oder ihn verlässt. ReplicationBytes ist der Verkehr, der der Datenreplikation gewidmet ist.

Weitere Informationen finden Sie unter [Netzwerkbezogene Grenzen](#).

- Komplexe Befehle: Redis-Befehle werden in einem einzigen Thread bereitgestellt, was bedeutet, dass Anfragen sequenziell 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 verwendet die „Big O“-Notation, 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 <https://redis.io/commands>.

Beispiele für mögliche Probleme:

- **Lua-Skripte:** Redis bietet einen eingebetteten Lua-Interpreter, der die Ausführung von Skripten auf der Serverseite ermöglicht. Lua-Skripte auf Redis 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, Datenparsen und andere direkt auf der Redis-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 eingeschalteter 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 Schlüsselmodifikationen in 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

erfordern erhebliche CPU-Zeit, wenn die Liste der Parameter groß ist oder eine große Liste, eine Menge, eine sortierte Menge oder einen Hash enthält (Datenstrukturen, die mehrere Unterelemente enthalten). Mit anderen Worten, selbst das Löschen eines einzelnen Schlüssels kann erhebliche Zeit in Anspruch nehmen, wenn es viele Elemente enthält. Die Alternative zu `DEL` ist `UNLINK`, ein asynchroner Befehl, der seit Redis 4 verfügbar ist. `UNLINK` hat Vorrang vor `DEL` wann immer möglich. Ab ElastiCache Redis 6.x verhält sich der `DEL` Befehl aufgrund des `lazy-free-lazy-user-del` Parameters so, als ob er aktiviert wäre `UNLINK`. Weitere Informationen finden Sie unter [Redis 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 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-Auslastung aus.

Obwohl die CPU-Auslastung allein nicht die Ursache für Konnektivitätsprobleme ist, kann es zu viel Zeit für die Verarbeitung einzelner oder einiger Befehle über mehrere Schlüssel zu einem Ausfall anderer Anforderungen führen und die CPU-Gesamtauslastung erhöhen.

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 wirken: Redis bietet auch Befehle, die unabhängig von ihrem Datentyp auf einen oder mehrere Schlüssel wirken. ElastiCache for Redis 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 andere Datenstrukturen CPU-intensiv sein. 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 <https://redis.io/commands>.

Netzwerkonnektivität

Nach der Überprüfung der Netzwerkkonfigurationen im Zusammenhang mit DNS-Auflösung, Sicherheitsgruppen, Netzwerk-ACLs und Routen-Tabellen kann die Konnektivität mit dem VPC Reachability Analyzer und den Systemtools überprüft werden.

Reachability Analyzer testet die Netzwerkonnektivität 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 Ihrer VPC verfügbaren ElastiCache Knoten. Gehen Sie dazu wie folgt vor:

1. Wechseln 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 oder auf andere Weise speichern Sie die ENI ID. 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. Erstellen Sie einen Analysepfad unter [https://console.aws.amazon.com/vpc/home? # ReachabilityAnalyzer](https://console.aws.amazon.com/vpc/home?#ReachabilityAnalyzer) und wählen Sie die folgenden Optionen aus:
 - Quelltyp: Wählen Sie die Instance aus, wenn Ihr ElastiCache Client auf einer Amazon EC2 EC2-Instance läuft, oder eine Netzwerkschnittstelle (falls er einen anderen Service verwendet, z. B. AWS Fargate Amazon ECS mit awsvpc-Netzwerk usw.) AWS Lambda, und die entsprechende Ressourcen-ID (EC2-Instance oder ENI-ID);
 - Typ Ziel: Wählen Sie aus Netzwerkschnittstelle und wählen Sie das Kontrollkästchen Elasticache ENI aus der Liste.
 - Zielport: Geben Sie 6379 für Redis oder 11211 ElastiCache für Memcached an. ElastiCache Dies sind die Ports, die mit der Standardkonfiguration definiert sind, und in diesem Beispiel wird davon ausgegangen, dass sie nicht geändert werden.
 - Protocol (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 bereitstellen. 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 schlägt fehl und die VPC Reachability Analyzer-Tests bestanden haben, bitten Sie Ihren Systemadministrator, mögliche hostbasierte Firewall-Regeln, asymmetrische Routingregeln oder andere mögliche Einschränkungen auf Betriebssystemebene zu überprüfen.

Überprüfen Sie auf der ElastiCache Konsole, ob die Verschlüsselung während der Übertragung in Ihren ElastiCache Clusterdetails aktiviert ist. Wenn die Verschlüsselung bei der Übertragung aktiviert ist, bestätigen Sie, ob die TLS-Sitzung mit folgendem Befehl eingerichtet werden kann:

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

Eine umfangreiche Ausgabe wird erwartet, wenn die Verbindung und TLS-Verhandlung erfolgreich sind. 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;
- Anwendung unterstützt TLS/SSL, verfügt aber nicht über die richtigen Konfigurationsflags oder vertrauenswürdige 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 `CurrConnections` 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](#), 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 die Plateaus über längere Zeiträume beobachtet werden, folgen sie wahrscheinlich Replikationsverzögerung, Zunahme der Bytes, die für den Cache verwendet werden, Drop auf freien Speicher, hohe Swap und CPU-Auslastung. Amazon-EC2-Instances verfügen auch über Netzwerklimits, die über [ENA-Treibermetriken](#) verfolgt werden können. Linux-Instances mit erweiterter Netzwerkunterstützung und ENA-Treibern 2.2.10 oder höher können die Limitindikatoren mit folgendem Befehl überprüfen:

```
# ethtool -S eth0 | grep "allowance_exceeded"
```


CPU-Verwendung

Die Metrik zur CPU-Auslastung ist der Ausgangspunkt der Untersuchung, und die folgenden Punkte können helfen, mögliche Probleme ElastiCache nebenbei einzugrenzen:

- **Redis 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 obige Ereignis geschah am 26. Dezember um 19:26:07 UTC, dauerte 4,8 Sekunden (4,823 ms) und wurde durch die `KEYS` Befehl vom Client 1.2.3.4 angefordert.

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 unter Windows mit: PowerShell

```
PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')
```

```

DateTime      : 12/26/2020 7:26:07 PM
UtcDateTime   : 12/26/2020 7:26:07 PM
LocalDateTime : 12/26/2020 2:26:07 PM
Date          : 12/26/2020 12:00:00 AM
Day           : 26
DayOfWeek     : Saturday
DayOfYear     : 361
Hour          : 19
Millisecond   : 0
Minute        : 26
Month         : 12
Offset        : 00:00:00Ticks           : 637446075670000000
UtcTicks      : 637446075670000000
TimeOfDay     : 19:26:07
Year          : 2020

```

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 Befehle mit O (1) -Zeitkomplexität häufig gemeldet werden, überprüfen Sie die anderen Faktoren für eine hohe CPU-Auslastung, die zuvor erwähnt wurden.

- **Latenzmetriken:** ElastiCache for Redis 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;
- **Redis-Laufzeitbefehle:**
 - **info commandstats:** Bietet eine Liste der seit dem Start der Redis-Engine ausgeführten Befehle, ihre kumulative Ausführungsnummer, 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:** Verwenden Sie ElastiCache für Redis-Versionen vor 2.8.22 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 2.8.22 wurde eine forkless-gestützte Backup- und Replikationsmethode eingeführt. AWS Die neue Methode kann Schreibvorgänge verzögern, um Fehler zu vermeiden. Beide Methoden können Perioden höherer CPU-Auslastung verursachen, zu höheren Reaktionszeiten führen und somit zu Client-Timeouts 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 Standardeinstellung ElastiCache für die Redis-Konfiguration 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 in diesem Zeitraum keine Anforderung sendet, beendet die Redis-Engine die Verbindung, sobald die Verbindung den Timeout-Wert erreicht. Kleine Zeitüberschreitungen 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 Standardeinstellung ElastiCache für Redis-Konfigurationen schränkt 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 kann Redis entscheiden, ElastiCache Clients zu trennen, die große Client-Ausgabepuffer verbrauchen, um Speicherplatz freizugeben und die Integrität 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 EC2 Instances

Die Auslastung und Reaktionsfähigkeit auf der Client-Seite können sich auch auf die Anfragen auswirken. ElastiCache EC2-Instance- und Betriebssystemlimits müssen sorgfältig überprüft werden, während zeitweilige Konnektivitäts- oder Timeout-Probleme behoben werden. Einige wichtige Punkte zu beachten:

- CPU:
 - EC2-Instance CPU-Auslastung: Stellen Sie sicher, dass die CPU nicht gesättigt ist oder fast 100 Prozent beträgt. Historische Analysen können wie folgt durchgeführt werden. Beachten Sie jedoch CloudWatch, dass die Granularität der Datenpunkte entweder 1 Minute (bei aktivierter detaillierter Überwachung) oder 5 Minuten beträgt;

- Bei Verwendung von [Burstable EC2-Instances](#), stellen Sie sicher, dass ihr CPU-Guthaben nicht erschöpft wurde. Diese Informationen sind in der `CPUcreditBalance` CloudWatch Metrik verfügbar.
- 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`.
- Network (Netzwerk)
 - Überprüfen Sie, ob der Netzwerkdurchsatz gemäß den Instance-Funktionen unter akzeptablen Werten liegt. Weitere Informationen finden Sie unter [Amazon-EC2-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-Meta-Daten gelöscht wurden, NTP über 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 von sehr kleinen Anfragen pro Sekunde besteht. ElastiCache Der Datenverkehr kann so optimiert werden, dass Netzwerkpakete besser genutzt werden können, und zwar über Pipelines oder Befehle, die beispielsweise mehrere Operationen gleichzeitig ausführen, 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
```

Aus der obigen Ausgabe können wir bestätigen, dass der TCP-3-Wege-Handshake in 222 Mikrosekunden (918091 - 917869) abgeschlossen wurde und der Ping-Befehl in 173 Mikrosekunden (918295 - 918122) 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. Verwenden des gleichen `RedisPING` Befehl mit `strace` wir bekommen:

```
$ 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"], 0x7ffffde7cc38 /* 22 vars */) = 0
1609430221.708955 (+ 0.011231) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|
SOCK_NONBLOCK, 0) = 3
1609430221.709084
(+ 0.000124) socket(AF_UNIX, SOCK_STREAM|SOCK_CLOEXEC|SOCK_NONBLOCK, 0) = 3
1609430221.709258 (+ 0.000173) open("/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709637 (+ 0.000378) open("/etc/host.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.709923
(+ 0.000286) open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
1609430221.711365 (+ 0.001443) open("/etc/hosts", O_RDONLY|O_CLOEXEC) = 3
1609430221.713293 (+ 0.001928) socket(AF_INET, SOCK_DGRAM|SOCK_CLOEXEC|SOCK_NONBLOCK,
IPPROTO_IP) = 3
1609430221.717419
(+ 0.004126) recvfrom(3, "\362|
\201\200\0\1\0\2\0\0\0\0\rnotls20201224\6tihew"..., 2048, 0, {sa_family=AF_INET,
sin_port=htons(53), sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 155
1609430221.717890 (+ 0.000469) recvfrom(3,
"\204\207\201\200\0\1\0\1\0\0\0\0\rnotls20201224\6tihew"...,
65536, 0, {sa_family=AF_INET, sin_port=htons(53),
sin_addr=inet_addr("172.31.0.2")}, [28->16]) = 139
1609430221.745659 (+ 0.027772) socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) = 3
```



```
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
    0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
    [128]) = -1 ENOTSOCK
    (Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
    [128->0]) = 7
+PONG
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
    (+ 0.003569) +++ exited with 0 +++
```

Im obigen Beispiel dauerte der Befehl etwas mehr als 54 Millisekunden ($752110 - 697712 = 54398$ Mikrosekunden).

Eine beträchtliche Zeit, etwa 20 ms, wurde benötigt, um nc zu instanzieren und die Namensauflösung durchzuführen (von 697712 bis 717890), danach waren 2 ms erforderlich, um den TCP-Socket zu erstellen (745659 bis 747858) und 0,4 ms (747858 bis 748330), um die Antwort für die Anfrage zu senden und zu erhalten.

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 in Amazon ElastiCache](#)
- [Richtlinie für den Datenverkehr zwischen Netzwerken](#)
- [Identity and Access Management für Amazon ElastiCache](#)
- [Konformitätsvalidierung für Amazon ElastiCache](#)
- [Ausfallsicherheit in Amazon ElastiCache](#)
- [Sicherheit der Infrastruktur in AWS ElastiCache](#)
- [Service-Updates in ElastiCache](#)
- [Common Vulnerabilities and Exposures \(CVE\): Sicherheitslücken, die in ElastiCache für Redis behoben wurden](#)

Datenschutz in Amazon ElastiCache

Das AWS-[Modell der geteilten Verantwortung](#) gilt für den Datenschutz in AWS ElastiCache (ElastiCache). Wie in diesem Modell beschrieben, ist AWS für den Schutz der globalen Infrastruktur verantwortlich, auf der die gesamte AWS-Cloud läuft. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Dieser Inhalt enthält die Sicherheitskonfigurations- und Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#).

Wir empfehlen aus Gründen des Datenschutzes, dass Sie AWS-Anmeldeinformationen schützen und die Konten jeweils mit AWS Identity and Access Management (IAM) einrichten. 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 die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie TLS für die Kommunikation mit AWS-Ressourcen.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- 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-Services über die Konsole, die API, AWS CLI, AWS oder die SDKs arbeiten. Alle Daten, die Sie in ElastiCache oder andere Services eingeben, können in die Diagnoseprotokolle aufgenommen werden. Wenn Sie eine URL für einen externen Server bereitstellen, schließen Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL ein.

Themen

- [Datensicherheit in Amazon ElastiCache](#)

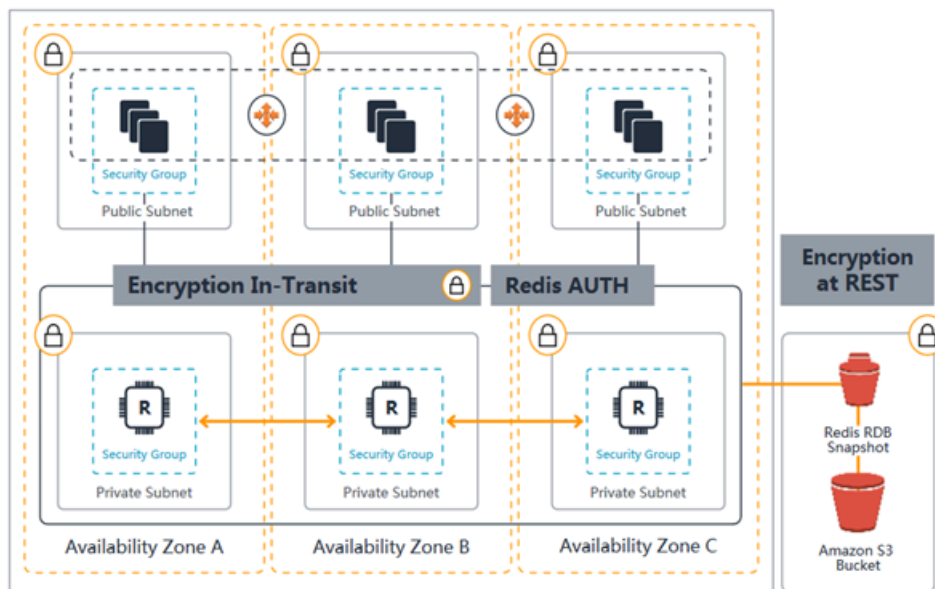
Datensicherheit in Amazon ElastiCache

Zur Gewährleistung der Sicherheit Ihrer Daten bieten Amazon ElastiCache und Amazon EC2 Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten auf dem Server.

Amazon ElastiCache für Redis bietet Verschlüsselungsfunktionen für Daten in Caches mit den Redis-Versionen 3.2.6 (für EOL geplant, siehe [Zeitplan für das Ende der Lebensdauer der Redis-Versionen](#)), 4.0.10 oder höher:

- 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.

Amazon ElastiCache für Redis unterstützt außerdem die Benutzerauthentifizierung entweder mithilfe von IAM und dem AUTH-Befehl von Redis sowie die Autorisierung von Benutzeroperationen mithilfe von Role-Based Access Control (RBAC).



ElastiCache-for-Redis-Sicherheitsdiagramm

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, bieten Amazon ElastiCache und Amazon EC2 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.

Für alle Serverless-Caches ist die Verschlüsselung während der Übertragung aktiviert. Für selbst entworfene Cluster können Sie die Verschlüsselung während der Übertragung für eine Replikationsgruppe aktivieren, indem Sie den Parameter `TransitEncryptionEnabled` bei der Erstellung der Replikationsgruppe auf `true` (CLI: `--transit-encryption-enabled`) festlegen. Sie können dies unabhängig davon tun, ob Sie die Replikationsgruppe mithilfe der AWS Management Console AWS CLI, der oder der ElastiCache API erstellen.

Themen

- [Übersicht über die Verschlüsselung während der Übertragung](#)
- [Bedingungen für die Verschlüsselung während der Übertragung](#)
- [Bewährte Methoden für die Verschlüsselung während der Übertragung](#)
- [Weitere Informationen finden Sie auch unter](#)
- [Aktivieren von Verschlüsselung während der Übertragung](#)
- [Herstellen einer Verbindung mit Amazon ElastiCache für Redis mit Verschlüsselung während der Übertragung mithilfe von redis-cli](#)
- [Aktivieren der Verschlüsselung während der Übertragung für einen selbst entworfenen Redis-Cluster mit Python](#)
- [Bewährte Methoden für die Aktivierung der Verschlüsselung während der Übertragung](#)

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 Client-Verbindungen – Client-Verbindungen mit 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.
- Clientauthentifizierung – Mit der AUTH-Funktion von Redis können Server die Clients authentifizieren.

Bedingungen für die Verschlüsselung während der Übertragung

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 für Replikationsgruppen unterstützt, die Redis-Versionen 3.2.6, 4.0.10 und höher.
- Das Ändern der Einstellung für die Verschlüsselung während der Übertragung für einen vorhandenen Cluster wird für Replikationsgruppen unterstützt, auf denen Redis Version 7 und höher ausgeführt wird.
- Die Verschlüsselung während der Übertragung wird nur für Replikationsgruppen unterstützt, die in einer 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 diese 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.

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 der Aufbau neuer Verbindungen kostspielig sein kann, können Sie die Auswirkungen der Verschlüsselung während der Übertragung auf die Leistung verringern, indem Sie Ihre TLS-Verbindungen aufrechterhalten.

Weitere Informationen finden Sie auch unter

- [Verschlüsselung im Ruhezustand in ElastiCache](#)
- [Authentifizieren mit dem Redis-AUTH-Befehl](#)
- [Authentifizierung von Benutzern mit rollenbasierter Zugriffskontrolle \(RBAC\)](#)
- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Identity and Access Management für Amazon ElastiCache](#)

Aktivieren von Verschlüsselung während der Übertragung

Für alle Serverless-Caches ist die Verschlüsselung während der Übertragung aktiviert. Für einen selbst entworfenen Cluster können Sie die Verschlüsselung während der Übertragung mit der AWS Management Console, der AWS-CLI oder der ElastiCache-API aktivieren.

Aktivieren der Verschlüsselung während der Übertragung mit der AWS Management Console

Aktivieren der Verschlüsselung während 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.

Eine schrittweise Vorgehensweise finden Sie unter folgender Quelle:

- [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Aktivieren 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 Übertragungsverschlüsselung auf `preferred` festlegen. In diesem Modus können Ihre Redis-Clients sowohl über verschlüsselte als auch über unverschlüsselte Verbindungen eine Verbindung herstellen. Nachdem Sie alle Ihre Redis-Clients auf verschlüsselte Verbindungen migriert haben, können Sie Ihre Clusterkonfiguration ändern, um den Modus der Verschlüsselung während der Übertragung auf `required` festzulegen. Wenn Sie den Modus der Verschlüsselung während der Übertragung auf `required` festlegen, werden alle unverschlüsselten Verbindungen gelöscht und nur verschlüsselte Verbindungen werden zugelassen.

Schritt 1: Ihren Transit encryption mode (Modus der Verschlüsselung während der Übertragung) auf Preferred (Bevorzugt) festlegen

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie Redis-Caches aus den Ressourcen für ElastiCache aus, die im Navigationsbereich auf der linken Seite aufgeführt sind.
3. Wählen Sie den Redis-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 der Migration aller Ihrer Redis-Clients auf verschlüsselte Verbindungen:

Schritt 2: Ihren Transit encryption mode (Modus der Verschlüsselung während der Übertragung) auf Required (Erforderlich) festlegen

1. Melden Sie sich in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie Redis-Caches aus den Ressourcen für ElastiCache aus, die im Navigationsbereich auf der linken Seite aufgeführt sind.
3. Wählen Sie den Redis-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 der Verschlüsselung während der Übertragung mit der AWS CLI

Wenn Sie die Verschlüsselung während der Übertragung beim Erstellen einer Redis-Replikationsgruppe mit der AWS CLI aktivieren möchten, verwenden Sie den `transit-encryption-enabled`-Parameter.

Aktivieren der Verschlüsselung während der Übertragung für neue selbst entworfene Redis-Cluster (Cluster-Modus deaktiviert) (CLI)

Verwenden Sie die AWS CLI-Operation `create-replication-group` und die folgenden Parameter, um eine Redis-Replikationsgruppe mit Replikaten zu erstellen, bei der die Verschlüsselung während der Übertragung aktiviert ist:

Hauptparameter:

- **--engine**—Der Wert muss `redis` sein.
- **--engine-version**—Muss 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:

- [Erstellen einer Redis-Replikationsgruppe \(Clustermodus deaktiviert\) \(AWS CLI\)](#)
- [create-replication-group](#)

Aktivieren der Verschlüsselung während der Übertragung für neue selbst entworfene Redis-Cluster (Cluster-Modus aktiviert) (CLI)

Verwenden Sie den AWS CLI-Vorgang `create-replication-group` und die folgenden Parameter, um eine Redis-Replikationsgruppe (Cluster-Modus aktiviert) zu erstellen, für die die Verschlüsselung während der Übertragung aktiviert ist:

Hauptparameter:

- **--engine**—Der Wert muss `redis` sein.
- **--engine-version**—Muss 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:

- [Erstellen einer Redis-Replikationsgruppe \(Clustermodus aktiviert\) von Grund auf \(AWS CLI\)](#)
- [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 Redis-Clients sowohl über verschlüsselte als auch über unverschlüsselte Verbindungen eine Verbindung herstellen. Nachdem Sie alle Ihre Redis-Clients auf verschlüsselte Verbindungen migriert haben, können Sie Ihre Clusterkonfiguration ändern, um den Modus der Verschlüsselung während der Übertragung auf `required` festzulegen. 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 Redis-Replikationsgruppe (Cluster-Modus aktiviert) zu aktualisieren, für die die Verschlüsselung während der Übertragung deaktiviert ist:

So aktivieren Sie die Verschlüsselung während der Übertragung

1. Den Modus der Verschlüsselung während der Übertragung mit den folgenden Parametern auf `preferred` festlegen
 - **--transit-encryption-enabled**—Erforderlich.
 - **--transit-encryption-mode** – muss auf `preferred` festgelegt sein.
2. Den Modus der Verschlüsselung während der Übertragung mit den folgenden Parametern auf `required` festlegen:
 - **--transit-encryption-enabled**—Erforderlich.
 - **--transit-encryption-mode** – muss auf `required` festgelegt sein.

Herstellen einer Verbindung mit Amazon ElastiCache für Redis mit Verschlüsselung während der Übertragung mithilfe von redis-cli

Um auf Daten aus ElastiCache für Redis-Caches zuzugreifen, die mit Verschlüsselung während der Übertragung aktiviert sind, verwenden Sie Clients, die mit Secure Socket Layer (SSL) arbeiten. Sie können redis-cli auch mit TLS/SSL in Amazon Linux und Amazon Linux 2 verwenden. Wenn Ihr Client TLS nicht unterstützt, können Sie den `stunnel`-Befehl auf Ihrem Client-Host verwenden, um einen SSL-Tunnel zu den Redis-Knoten zu erstellen.

Verschlüsselte Verbindung mit Linux

Gehen Sie folgendermaßen vor, um mit redis-cli eine Verbindung zu einem Redis-Cluster herzustellen, der mit Verschlüsselung während der Übertragung auf Amazon Linux 2023, Amazon Linux 2 oder Amazon Linux aktiviert ist.

1. Laden Sie das redis-cli-Dienstprogramm herunter und kompilieren Sie es. Dieses Dienstprogramm ist Teil der Redis-Softwaredistribution.
2. Geben Sie an der Eingabeaufforderung Ihrer EC2-Instance 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 dabei den Endpunkt Ihres Clusters und den Port durch das, was in diesem Beispiel gezeigt wird.

```
redis-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 http://download.redis.io/redis-stable.tar.gz
```

```
tar xvzf redis-stable.tar.gz
cd redis-stable
make distclean
make redis-cli BUILD_TLS=yes
sudo install -m 755 src/redis-cli /usr/local/bin/
```

Amazon Linux

Wenn Sie Amazon Linux verwenden, geben Sie Folgendes ein:

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

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 redis-cli-Dienstprogramm heruntergeladen und installiert haben, wird empfohlen, den optionalen make-test Befehl auszuführen.
4. Um eine Verbindung zu einem Cluster mit aktivierter Verschlüsselung und Authentifizierung herzustellen, geben Sie diesen Befehl ein:

```
redis-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 Befehl `redis6-cli` anstelle von `redis-cli` verwenden:

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

Verschlüsselte Verbindung mit Stunnel

Gehen Sie wie folgt vor, um mit `redis-cli` eine Verbindung zu einem Redis-Cluster herzustellen, der mit Verschlüsselung während der Übertragung über `stunnel` aktiviert ist.

1. Verwenden Sie SSH, um eine Verbindung mit Ihrem Client herzustellen und installieren Sie `stunnel`.

```
sudo yum install stunnel
```

2. Führen Sie den folgenden Befehl aus, um eine Datei `/etc/stunnel/redis-cli.conf` gleichzeitig zu erstellen und zu bearbeiten, um einen ElastiCache für Redis-Cluster-Endpunkt zu einem oder mehreren Verbindungsparametern hinzuzufügen, wobei Sie die unten angegebene Ausgabe als Vorlage verwenden.

```
vi /etc/stunnel/redis-cli.conf

fips = no
setuid = root
setgid = root
pid = /var/run/stunnel.pid
debug = 7
delay = yes
options = NO_SSLv2
options = NO_SSLv3
[redis-cli]
  client = yes
  accept = 127.0.0.1:6379
  connect = primary.ssltest.wif01h.use1.cache.amazonaws.com:6379
[redis-cli-replica]
  client = yes
  accept = 127.0.0.1:6380
  connect = ssltest-02.ssltest.wif01h.use1.cache.amazonaws.com:6379
```

In diesem Beispiel hat die Konfigurationsdatei zwei Verbindungen: `redis-cli` und `redis-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 Redis-Standard 127.0.0.1 an Port 6379 eingestellt. 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 [Temporäre Ports](#) im Benutzerhandbuch zu Amazon VPC.

- Verbinden ist auf den Endpunkt des Redis-Servers festgelegt. Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#).

3. Starten stunnel.

```
sudo stunnel /etc/stunnel/redis-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 eine Verbindung zum verschlüsselten Redis-Knoten unter Verwendung des lokalen Endpunkts des Tunnels her.

- Wenn während ElastiCache der Erstellung des Redis-Clusters kein AUTH-Passwort verwendet wurde, verwendet dieses Beispiel das `redis-cli`, um eine Verbindung mit dem ElastiCache für Redis-Server unter Verwendung des vollständigen Pfads für `redis-cli` unter Amazon Linux herzustellen:

```
/home/ec2-user/redis-stable/src/redis-cli -h localhost -p 6379
```

Wenn bei der Erstellung des Redis-Clusters das AUTH-Passwort verwendet wurde, wird in diesem Beispiel `redis-cli` verwendet, um sich mit dem Redis-Server zu verbinden, wobei der vollständige Pfad für `redis-cli` auf Amazon Linux verwendet wird:

```
/home/ec2-user/redis-stable/src/redis-cli -h localhost -p 6379 -a my-secret-password
```

ODER

- Ändern Sie das Verzeichnis in `redis-stable`, und führen Sie die folgenden Schritte aus:

Wenn während ElastiCache der Erstellung des Redis-Clusters kein AUTH-Passwort verwendet wurde, verwendet dieses Beispiel das `redis-cli`, um eine Verbindung mit dem ElastiCache für Redis-Server unter Verwendung des vollständigen Pfads für `redis-cli` unter Amazon Linux herzustellen:

```
src/redis-cli -h localhost -p 6379
```

Wenn bei der Erstellung des Redis-Clusters das AUTH-Passwort verwendet wurde, wird in diesem Beispiel `redis-cli` verwendet, um sich mit dem Redis-Server zu verbinden, wobei der vollständige Pfad für `redis-cli` auf Amazon Linux verwendet wird:

```
src/redis-cli -h localhost -p 6379 -a my-secret-password
```

Dieses Beispiel verwendet `Telnet`, um eine Verbindung zum Redis-Server herzustellen.

```
telnet localhost 6379

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
auth MySecretPassword
+OK
get foo
$3
bar
```

5. Um SSL-Tunnel zu stoppen und zu schließen, verwenden Sie `kill` für den `stunnel`-Prozess.

```
sudo kill stunnel
```

Aktivieren der Verschlüsselung während der Übertragung für einen selbst entworfenen Redis-Cluster mit Python

Die folgende Anleitung zeigt, wie die Verschlüsselung während der Übertragung auf einem Cluster von Redis 7.0 aktiviert wird, der ursprünglich mit deaktivierter Verschlüsselung während der Übertragung erstellt wurde. TCP- und TLS-Clients kommunizieren während dieses Vorgangs ohne Ausfallzeiten weiter mit dem Cluster.

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 Beispiel unten wurde über eine EC2-Instance verarbeitet, die in derselben VPC gestartet wurde, mit der der ElastiCache-Redis-Cluster erstellt wird.

Note

- In den folgenden Beispielen wird das boto3-SDK-für-ElastiCache-Verwaltungsvorgänge (Cluster- oder Benutzererstellung) und `redis-py/redis-py-cluster` für die Datenverarbeitung verwendet.
- Sie müssen mindestens boto3-Version (=~) 1.26.39 verwenden, um die Online-TLS-Migration mit der Cluster-Änderungs-API verwenden zu können.
- ElastiCache unterstützt die Online-TLS-Migration nur für Redis-Cluster mit Version 7.0 oder höher. Wenn Sie also einen Cluster haben, auf dem eine ältere Redis-Version als 7.0 ausgeführt wird, müssen Sie die Redis-Version Ihres Clusters aktualisieren. Weitere Informationen zu Versionsunterschieden finden Sie unter [Wesentliche Versionsverhalten und Kompatibilitätsunterschiede](#).

Themen

- [Definieren der Zeichenfolgenkonstanten, die den ElastiCache-Redis-Cluster starten](#)
- [Definieren der Klassen für die Clusterkonfiguration](#)
- [Definieren einer Klasse, die den eigentlichen Cluster darstellt](#)
- [\(Optional\) Erstellen einer Wrapper-Klasse, um die Client-Verbindung mit dem Redis-Cluster zu demonstrieren](#)
- [Erstellen der Hauptfunktion, die den Änderungsprozess der Konfiguration der Verschlüsselung während der Übertragung demonstriert](#)

Definieren der Zeichenfolgenkonstanten, die den ElastiCache-Redis-Cluster starten

Zunächst definieren wir einige einfache Python-Zeichenkettenkonstanten, die die Namen der AWS-Entitäten enthalten, die für die Erstellung des ElastiCache-Clusters erforderlich sind, z. B. `security-group`, `Cache Subnet group` und `default parameter group`. Alle diese

AWS-Entitäten müssen vorab 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

Jetzt definieren wir einige einfache Python-Klassen, die eine Konfiguration eines Clusters darstellen und Metadaten über den Cluster enthalten, z. B. die Redis-Version, den Instance-Typ und die Angabe, ob die Verschlüsselung während der Übertragung (TLS) aktiviert oder deaktiviert ist.

```
#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

Jetzt definieren wir einige einfache Python-Klassen, die den eigentlichen ElastiCache-Redis-Cluster darstellen. Diese Klasse enthält ein Client-Feld mit einem Boto3-Client für ElastiCache-Verwaltungsvorgänge wie das Erstellen des Clusters und das Abfragen der 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 einer Wrapper-Klasse, um die Client-Verbindung mit dem Redis-Cluster zu demonstrieren

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-ElastiCache-Client.
2. Initialisieren Sie den `redis-py-cluster`-Client, der eine Verbindung mit dem Cluster mit einer guten TCP-Verbindung ohne TLS herstellt.
3. Der `redis-py-cluster`-Client füllt den Cluster vorab mit einigen Daten.
4. Der boto3-Client löst die TLS-Migration von „no-TLS“ zu „TLS Preferred“ aus.
5. Während der Cluster zu TLS Preferred migriert wird, sendet der `redis-py-cluster-TCP`-Client wiederholte `get`-Operationen an den Cluster, bis die Migration abgeschlossen ist.
6. Nachdem die Migration zu TLS Preferred abgeschlossen ist, überprüfen wir, dass der Cluster die Verschlüsselung während der Übertragung unterstützt. Anschließend erstellen wir einen `redis-py-cluster`-Client, der mit TLS eine Verbindung mit dem Cluster herstellt.
7. Wir senden einige `get`-Befehle mit dem neuen TLS-Client und dem alten TCP-Client.
8. Der boto3-Client löst die TLS-Migration von „TLS Preferred“ zu „TLS required“ aus.
9. Während der Cluster zu „TLS required“ migriert wird, 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

Vor dem Aktivieren der Verschlüsselung während der Übertragung: sicherstellen, dass die DNS-Einträge ordnungsgemäß verarbeitet werden

Note

Während dieses Prozesses ändern und löschen wir alte Endpunkte. Eine falsche Verwendung der Endpunkte kann dazu führen, dass der Redis-Client alte und gelöschte Endpunkte verwendet, sodass er keine Verbindung zum Cluster herstellen kann.

Während der Cluster von „no-TLS“ zu „TLS Preferred“ migriert wird, werden die alten DNS-Einträge pro Knoten beibehalten und die neuen DNS-Einträge pro Knoten werden in einem anderen Format generiert. TLS-fähige Cluster verwenden ein anderes Format für DNS-Einträge als nicht TLS-fähige Cluster. ElastiCache speichert beide DNS-Einträge, wenn ein Cluster im Verschlüsselungsmodus konfiguriert ist: bevorzugt, damit Anwendungen und andere Redis-Clients zwischen ihnen wechseln können. Die folgenden Änderungen in den DNS-Einträgen finden während des TLS-Migrationsprozesses statt:

Beschreibung der Änderungen an den DNS-Einträgen, die bei der Aktivierung der Verschlüsselung während der Übertragung vorgenommen werden

Für CME-Cluster

Wenn ein Cluster auf „Übertragungs-Verschlüsselungsmodus: bevorzugt“ eingestellt ist:

- Die ursprünglichen Cluster-Endpunkte für nicht TLS-fähige Cluster bleiben aktiv. Es wird keine Ausfallzeit geben, wenn der Cluster vom TLS-Verschlüsselungsmodus „none“ auf „preferred“ umkonfiguriert wird.
- Neue TLS-Redis-Endpunkte werden generiert, wenn der Cluster auf den Modus „TLS Preferred“ eingestellt ist. Diese neuen Endpunkte werden auf dieselben IPs wie die alten (ohne TLS) aufgelöst.

- Der neue TLS-Redis-Konfigurationsendpunkt wird in der ElastiCache-Konsole und in der Antwort auf die `describe-replication-group`-API verfügbar gemacht.

Wenn ein Cluster auf „Übertragungs-Verschlüsselungsmodus: erforderlich“ eingestellt ist:

- Alte Endpunkte, die nicht TLS-fähig sind, werden gelöscht. Es wird keine Ausfallzeiten der TLS-Cluster-Endpunkte geben.
- Sie können einen neuen `cluster-configuration-endpoint` von der ElastiCache-Konsole oder von der `describe-replication-group`-API abrufen.

Für CMD-Cluster mit aktiviertem automatischem Failover oder deaktiviertem automatischen Failover

Wenn die Replikationsgruppe auf „Übertragungs-Verschlüsselungsmodus: bevorzugt“ eingestellt ist:

- Der ursprüngliche primäre Endpunkt und der Reader-Endpunkt für nicht TLS-fähige Cluster bleiben aktiv.
- Neue primäre und Reader-Endpunkte mit TLS werden generiert, wenn der Cluster auf den Modus „TLS Preferred“ eingestellt ist. Diese neuen Endpunkte werden auf dieselben IPs wie die alten (ohne TLS) aufgelöst.
- Der neue primäre Endpunkt und der Reader-Endpunkt werden in der ElastiCache-Konsole und in der Antwort auf die `describe-replication-group`-API verfügbar gemacht.

Wenn die Replikationsgruppe auf „Übertragungs-Verschlüsselungsmodus: erforderlich“ eingestellt ist:

- Der ursprüngliche primäre Endpunkt und der Reader-Endpunkt für nicht TLS-fähige Cluster bleiben aktiv.
- Alte primäre und Reader-Endpunkte ohne TLS werden gelöscht. Es wird keine Ausfallzeiten der TLS-Cluster-Endpunkte geben.
- Sie können neue primäre und Reader-Endpunkte von der ElastiCache-Konsole oder von der `describe-replication-group` API abrufen.

Die empfohlene Verwendung der DNS-Einträge

Für CME-Cluster

- Verwenden Sie den Endpunkt der Clusterkonfiguration anstelle von DNS-Einträgen pro Knoten in Ihrem Anwendungscode. Es wird nicht empfohlen, DNS-Namen pro Knoten direkt zu verwenden, da sie sich ändern können, wenn Shards hinzugefügt oder entfernt werden.
- 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. Nachdem die Verschlüsselung während der Übertragung abgeschlossen ist, fragen Sie den Endpunkt der Clusterkonfiguration mit der `describe-replication-group`-API ab (wie oben gezeigt (fett gedruckt)) und verwenden Sie das DNS, das Sie ab diesem Zeitpunkt als Antwort erhalten.

Für CMD-Cluster mit aktiviertem automatischen Failover

- Verwenden Sie in Ihrem Anwendungscode den primären Endpunkt und den Reader-Endpunkt anstelle der DNS-Namen pro Knoten, da die alten DNS-Namen pro Knoten gelöscht und neue generiert werden, wenn der Cluster von „no-TLS“ zu „TLS Preferred“ migriert wird. Es wird nicht empfohlen, DNS-Namen pro Knoten direkt zu verwenden, da Sie Ihrem Cluster später möglicherweise Replikate hinzufügen. Wenn automatisches Failover aktiviert ist, werden außerdem die Rollen des primären Clusters und der Replikate automatisch vom ElastiCache-Service geändert. Es wird empfohlen, den primären Endpunkt und den Reader-Endpunkt zu verwenden, damit Sie den Überblick über diese Änderungen behalten können. Schließlich hilft Ihnen die Verwendung des Reader-Endpunkts dabei, Ihre Lesevorgänge aus den Replikaten gleichmäßig auf die Replikate im Cluster zu verteilen.
- Es wird nicht empfohlen, den primären Endpunkt und den Reader-Endpunkt in Ihrer Anwendung fest zu kodieren, da diese während des TLS-Migrationsprozesses geändert werden können. Nachdem die Migration zu „TLS Preferred“ abgeschlossen ist, fragen Sie den primären Endpunkt und den Reader-Endpunkt mit der `describe-replication-group`-API ab und verwenden Sie das DNS, das Sie ab diesem Zeitpunkt 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 Reader-Endpunkt anstelle der DNS-Namen pro Knoten in Ihrem Anwendungscode. Wenn automatisches Failover deaktiviert ist, übernehmen Sie stattdessen Skalierung, Patching, Failover und andere Aufgaben, die automatisch vom ElastiCache-Service verwaltet werden, wenn automatisches Failover aktiviert ist. Das erleichtert

es Ihnen, den Überblick über die verschiedenen Endpunkte zu behalten. Da bei der Migration des Clusters von „no-TLS“ zu „TLS Preferred“ die alten DNS-Namen pro Knoten gelöscht und neue generiert werden, sollten Sie die DNS-Namen pro Knoten nicht direkt verwenden. Dies ist erforderlich, damit sich die Clients während der TLS-Migration mit dem Cluster verbinden können. Außerdem profitieren Sie von einer gleichmäßigen Verteilung der Lesevorgänge zwischen den Replikaten, wenn Sie den Reader-Endpunkt verwenden, und behalten den Überblick über die DNS-Einträge, wenn Sie dem Cluster Replikate hinzufügen oder löschen.

- Eine feste Kodierung des Endpunkts der Clusterkonfiguration in Ihrer Anwendung wird nicht empfohlen, da er während der TLS-Migration geändert werden kann.

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 sowohl TCP- als auch TLS-Verbindungen akzeptieren und bereitstellen. Daher sollten Sie keine Clients erstellen, die versuchen, TLS-Verbindungen mit dem 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 des SNS-Services, um benachrichtigt zu werden, wenn die Verschlüsselung abgeschlossen ist
- Verwenden der `describe-events`-API, die ein Ereignis ausgibt, wenn die Verschlüsselung abgeschlossen ist
- Anzeige einer Meldung in der ElastiCache-Konsole, 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“)

- Bestätigen Sie, dass für den Cluster `transit_encryption_enabled` auf „Wahr“ festgelegt ist, indem Sie die `describe-replication-group`-API abfragen.

Nach dem Aktivieren der Verschlüsselung während der Übertragung: sicherstellen, dass die von Ihnen verwendeten Clients ordnungsgemäß konfiguriert sind

Während sich der Cluster im Modus „TLS Preferred“ befindet, sollte Ihre Anwendung TLS-Verbindungen mit dem Cluster herstellen und nur diese Verbindungen verwenden. Auf diese Weise kommt es bei Ihrer Anwendung nicht zu Ausfallzeiten, wenn die Verschlüsselung während der Übertragung aktiviert wird. Sie können sicherstellen, dass keine besseren TCP-Verbindungen mit der Redis-Engine bestehen, indem Sie den Befehl „Redis info“ im Abschnitt „SSL“ verwenden.

```
# 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
```

Verschlüsselung im Ruhezustand in ElastiCache

Zur Gewährleistung der Sicherheit Ihrer Daten bieten Amazon ElastiCache und Amazon S3 verschiedene Möglichkeiten, den Zugriff auf die Daten in Ihrem Cache zu beschränken. Weitere Informationen erhalten Sie unter [Amazon VPCs und ElastiCache Sicherheit](#) und [Identity and Access Management für Amazon ElastiCache](#).

Die Verschlüsselung im Ruhezustand von ElastiCache ist eine Funktion zur Erhöhung der Datensicherheit durch Verschlüsselung der 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-Laufwerke) in Data-Tiering-fähigen Clustern gespeichert sind, werden immer verschlüsselt.

ElastiCache bietet eine standardmäßige (vom Service verwaltete) Verschlüsselung im Ruhezustand sowie die Möglichkeit, eigene symmetrische, vom Kunden verwaltete AWS-KMS-Schlüssel in [AWS Key Management Service \(KMS\)](#) zu verwenden. 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 (Service Managed) ist die einzige Option, die in den GovCloud-Regionen (US) verfügbar ist.

Important

Wenn Sie die Verschlüsselung im Ruhezustand auf einem vorhandenen selbst entworfenen Redis-Cluster aktivieren möchten, müssen Sie Ihre vorhandene Replikationsgruppe löschen, nachdem Sie das Backup und die 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](#)
- [Verwendung von kundenverwalteten Schlüsseln aus AWS-KMS](#)
- [Aktivieren der Verschlüsselung im Ruhezustand](#)
- [Weitere Informationen finden Sie unter:](#)

Bedingungen für die Verschlüsselung im Ruhezustand

Die folgenden Einschränkungen für die ElastiCache-Verschlüsselung im-Ruhezustand sollten bei der Planung der Implementierung beachtet werden:

- Verschlüsselung im Ruhezustand wird nur für Replikationsgruppen unterstützt, die Redis-Version 3.2.6 (für das EOL geplant, siehe [Zeitplan für das Ende der Lebensdauer der Redis-Versionen](#)), 4.0.10 oder höher verwenden.
- Die Verschlüsselung der Daten im Ruhezustand wird nur für Replikationsgruppen unterstützt, die in einer 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 auf SSDs gespeicherte Daten unabhängig davon verschlüsselt, ob die Verschlüsselung im Ruhezustand aktiviert ist oder nicht.
- Die Option, einen vom Kunden verwalteten Schlüssel für die Verschlüsselung im Ruhezustand zu verwenden, ist in den AWS-Regionen GovCloud (us-gov-east-1 und us-gov-west-1) nicht verfügbar.
- Wenn ein Cluster einen Knotentyp aus der R6gd-Familie verwendet, werden auf SSDs gespeicherte Daten mit dem gewählten vom Kunden verschlüsselten AWS KMS-Schlüssel (bzw. vom Service verwalteter Verschlüsselung in AWS GovCloud-Regionen) verschlüsselt.

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.

Verwendung von kundenverwalteten Schlüsseln aus AWS-KMS

ElastiCache unterstützt symmetrische vom Kunden verwaltete AWS-KMS-Schlüssel (KMS-Schlüssel) für die Verschlüsselung im Ruhezustand. Kundenverwaltete KMS-Schlüssel sind Verschlüsselungsschlüssel, die Sie erstellen, besitzen und in Ihrem AWS-Konto verwalten. Weitere Informationen finden Sie unter [AWS-KMS-Schlüssel](#) im Entwicklerhandbuch zum AWS-Schlüsselverwaltungsdienst. Die Schlüssel müssen in AWS KMS erstellt werden, bevor Sie in ElastiCache verwendet werden können.

Wie Sie AWS-KMS-Root-Schlüssel erstellen können, erfahren Sie unter [Erstellen von Schlüsseln](#) im AWS-Entwicklerhandbuch zum Schlüsselverwaltungsdienst.

ElastiCache ermöglicht Ihnen die Integration in AWS KMS. Weitere Informationen finden Sie unter [Verwendung von Berechtigungen](#) im Entwicklerhandbuch zum AWS-Schlüsselverwaltungsdienst. Es ist keine Kundenaktion notwendig, um eine Amazon-ElastiCache-Integration mit AWS-KMS zu aktivieren.

Der `kms:ViaService`-Bedingungsschlüssel beschränkt die Verwendung eines AWS-KMS-Schlüssels (KMS-Schlüssel) auf Anfragen von bestimmten AWS-Services. Um `kms:ViaService` mit ElastiCache zu verwenden, fügen Sie beide ViaService-Namen in den Schlüsselwert der Bedingung

ein: `elasticache.AWS_region.amazonaws.com` und `dax.AWS_region.amazonaws.com`. Weitere Informationen finden Sie unter [kms:ViaService](#).

Sie können [AWS-CloudTrail](#) verwenden, um die Anfragen zu verfolgen, die Amazon ElastiCache in Ihrem Namen an AWS Key Management Service sendet. Alle API-Aufrufe an AWS Key Management Service in Bezug auf kundenverwaltete CMKs haben entsprechende CloudTrail-Protokolle. Sie können auch die Genehmigungen, die ElastiCache erstellt, durch Aufrufen des [ListGrants](#)-KMS-API-Aufrufs anzeigen.

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.
- Standardmäßig werden manuell erstellte Sicherungen mit dem der Replikationsgruppe zugeordneten KMS-Schlüssel verschlüsselt. 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 wahlweise 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 Standardverschlüsselung unter Verwendung 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. Durch diese Option wird die in Amazon

S3 gespeicherte Sicherungsdatei mit einem KMS-Schlüssel verschlüsselt, auch wenn 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. Das bedeutet, dass er nach einem Hardwareausfall nicht geändert oder wiederhergestellt werden kann. 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-Root-Schlüssel erhalten, so dass die Rotation keinen Einfluss auf den Zugriff auf Ihre ElastiCache-Daten hat. Für verschlüsselte Amazon-ElastiCache-Caches ist keine manuelle Schlüsselrotation möglich, bei der ein neuer Stammschlüssel erstellt und alle Verweise auf den alten Schlüssel aktualisiert werden. Weitere Informationen finden Sie unter [Rotieren von AWS-KMS-Schlüsseln](#) im Entwicklerhandbuch zum AWS-Schlüsselverwaltungsdienst.
- Die Verschlüsselung eines ElastiCache-Caches unter Verwendung eines KMS-Schlüssels erfordert eine Genehmigung pro Cache. 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-Genehmigungen und Grenzwerten finden Sie unter [Grenzwerte](#) im Entwicklerhandbuch zum AWS-Schlüsselverwaltungsdienst.

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 nur beim Erstellen eines ElastiCache-Caches aktivieren. Verwenden Sie hierzu die AWS Management Console, den AWS CLI oder die ElastiCache-API.

Beim Erstellen eines Caches können Sie eine der folgenden Optionen auswählen:

- Standard – Diese Option verwendet serviceverwaltete Verschlüsselung im Ruhezustand.
- Kundenverwalteter Schlüssel – Mit dieser Option können Sie die Schlüssel-ID/ARN von AWS-KMS für die Verschlüsselung im Ruhezustand bereitstellen.

Wie Sie AWS-KMS-Root-Schlüssel erstellen können, erfahren Sie unter [Erstellen von Schlüsseln](#) im Entwicklerhandbuch zum AWS-Schlüsselverwaltungsdienst

Inhalt

- [Aktivieren der Verschlüsselung der Daten im Ruhezustand mit der AWS Management Console](#)
- [Aktivieren der Verschlüsselung der Daten im Ruhezustand mit der AWS CLI](#)

Aktivieren der Verschlüsselung im Ruhezustand für einen vorhandenen selbst entworfenen Redis-Cluster

Sie können die Verschlüsselung nur dann aktivieren, wenn Sie eine 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](#) oder [Löschen einer Replikationsgruppe](#).

Aktivieren der Verschlüsselung der Daten im Ruhezustand mit 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 zur Verschlüsselung der Daten verwendet. Legen Sie die folgenden Einstellungen fest, um Ihren eigenen AWS KMS-Schlüssel auszuwählen:

- 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 unter der Einstellung Verschlüsselungsschlüssels die Option Kundenverwalteter CMK 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 die Option Standardschlüssel oder Kundenverwalteter CMK.

Das Schritt-für-Schritt-Verfahren finden Sie im Folgenden:

- [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Aktivieren der Verschlüsselung der Daten im Ruhezustand mit der AWS CLI

Verwenden Sie zum Erstellen einer Replikationsgruppe beim Erstellen eines Redis-Clusters mithilfe der AWS CLI den Parameter `--at-rest-encryption-enabled`, wenn Sie eine Replikationsgruppe erstellen.

Aktivieren der Verschlüsselung der Daten im Ruhezustand für einen Redis-Cluster (Cluster-Modus deaktiviert) (CLI)

Der folgende Vorgang erstellt die Redis-Replikationsgruppe (Cluster-Modus deaktiviert) `my-classic-rg` mit drei Knoten (`--num-cache-clusters`), einem primären und zwei Lesereplikaten. Die Verschlüsselung der Daten 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`**—Der Wert muss `redis` sein.
- **`--engine-version`**—Muss 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: In einem Redis-Cluster (Cluster-Modus 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:

- [Erstellen einer Redis-Replikationsgruppe \(Clustermodus deaktiviert\) \(AWS CLI\)](#)
- [create-replication-group](#)

Aktivieren der Verschlüsselung im Ruhezustand in einem Cluster für Redis (Cluster-Modus aktiviert) (CLI)

Die folgende Operation erstellt die Redis-Replikationsgruppe (Cluster-Modus 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 der Daten 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`—Der Wert muss `redis` sein.
- `--engine-version`—Muss 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 Redis-Cluster (Cluster-Modus 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 \  
  --at-rest-encryption-enabled
```

```
--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:

- [Erstellen einer Redis-Replikationsgruppe \(Clustermodus aktiviert\) von Grund auf \(AWS CLI\)](#)
- [create-replication-group](#)

Weitere Informationen finden Sie unter:

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Identity and Access Management für Amazon ElastiCache](#)

Authentifizierung und Autorisierung

ElastiCache unterstützt jetzt die Benutzerauthentifizierung mithilfe von IAM und dem AUTH-Befehl von Redis sowie die Autorisierung von Benutzeroperationen mithilfe von Role-Based Access Control (RBAC).

Themen

- [Rollenbasierte Zugriffskontrolle \(RBAC\)](#)
- [Authentifizieren mit dem Redis-AUTH-Befehl](#)
- [Deaktivieren der Zugriffskontrolle in einem Redis-Cache von ElastiCache](#)

Rollenbasierte Zugriffskontrolle (RBAC)

Anstatt Benutzer wie in [Authentifizieren mit dem Redis-AUTH-Befehl](#) beschrieben mit dem Redis-AUTH-Befehl zu authentifizieren, können Sie ab Redis 6.0 eine Funktion namens Role-Based Access Control (RBAC, rollenbasierte Zugriffskontrolle) verwenden. RBAC ist auch die einzige Möglichkeit, den Zugriff auf Serverless-Caches zu kontrollieren.

Im Unterschied zu Redis-AUTH, bei dem alle authentifizierten Clients vollen Zugriff auf den Cache haben, wenn ihr Token authentifiziert ist, können Sie mit RBAC den Cache-Zugriff über Benutzergruppen steuern. Diese Benutzergruppen dienen dazu, den Zugriff auf Caches zu organisieren.

Mithilfe von RBAC erstellen Sie Benutzer und weisen ihnen bestimmte Berechtigungen zu, indem Sie eine Zugriffszeichenfolge verwenden, wie im Folgenden beschrieben. Die Benutzer werden Benutzergruppen zugewiesen, die auf eine bestimmte Rolle ausgerichtet sind (Administratoren, Personalabteilung) und dann für einen oder mehrere Caches von ElastiCache für Redis bereitgestellt werden. Dadurch können Sie Sicherheitsgrenzen zwischen Clients einrichten, die den- bzw. dieselben Redis-Caches verwenden, und verhindern, dass Clients gegenseitig auf ihre Daten zugreifen.

RBAC wurde entwickelt, um die Einführung von [Redis ACL](#) in Redis 6 zu unterstützen. Wenn Sie RBAC mit Ihrem Cache von ElastiCache für Redis verwenden, gibt es einige Einschränkungen:

- Sie können keine Passwörter in einer Zugriffsfolge angeben. Sie legen Kennwörter mit den Aufrufen [CreateUser](#) oder [ModifyUser](#).
- 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 Kennwörter mit API-Parametern an und ElastiCache for Redis verwaltet Kennwörter. Daher können Sie `reset` nicht nutzen, da es alle Kennwörter für einen Benutzer entfernen würde.
- Redis 6 führt den [ACL-LISTE](#)-Befehl ein. Dieser Befehl gibt eine Liste der Benutzer zusammen mit den ACL-Regeln zurück, die auf jeden Benutzer angewendet wurden. ElastiCache for Redis unterstützt den `ACL LIST`-Befehl, enthält aber keinen Support für Passwort-Hashes wie Redis

sie hat. Mit ElastiCache for Redis können Sie die [describe-user](#)-Funktion verwenden, um ähnliche Informationen zu erhalten, einschließlich der Regeln, die in der Zugriffszeichenfolge enthalten sind. Jedoch ruft [describe-user](#) kein Benutzerkennwort ab.

Andere schreibgeschützte Befehle, die von ElastiCache for Redis unterstützt werden, sind [ACL WHOAMI](#), [ACL USERS](#), und [ACL CAT](#). ElastiCache for Redis unterstützt keinerlei andere schreibbasierte ACL-Befehle.

- Die folgenden Einschränkungen gelten:

| Ressource | Maximum erlaubt |
|-----------------------------|-----------------|
| Benutzer pro Benutzergruppe | 100 |
| Anzahl Benutzer | 1000 |
| Anzahl der Benutzergruppen | 100 |

Die Verwendung von RBAC mit ElastiCache for Redis wird im Folgenden detaillierter beschrieben.

Themen

- [Spezifizierung von Berechtigungen mithilfe einer Zugriffszeichenfolge](#)
- [Anwenden von RBAC auf einen Cache für ElastiCache für Redis](#)
- [Migrieren von Redis AUTH zu RBAC](#)
- [Migration von RBAC zu Redis-AUTH](#)
- [Automatisches Rotieren von Passwörtern für Benutzer](#)
- [Authentifizieren mit IAM](#)

Spezifizierung von Berechtigungen mithilfe einer Zugriffszeichenfolge

Wenn Sie Berechtigungen für einen Cache von ElastiCache für Redis angeben möchten, erstellen Sie eine Zugriffszeichenfolge und weisen sie einem Benutzer entweder mit der AWS CLI oder der AWS Management Console zu.

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.

Weitere Informationen 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 auf *command1* beschränkt.

`+@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 [Erstellen von Benutzern und Benutzergruppen mit der Konsole und dem CLI](#).

Wenn Sie einen vorhandenen Workload zu ElastiCache migrieren, können Sie die Zugriffszeichenfolge durch den Aufruf von `ACL LIST` abrufen, wobei der Benutzer und alle Kennwort-Hashes ausgeschlossen werden.

Für Redis Version 6.2 und höher wird auch die folgende Syntax für Zugriffszeichenfolgen unterstützt:

- `&*` – Zugriff auf alle verfügbaren Kanäle ist erlaubt.

Für Redis 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 [Wichtigen Berechtigungen](#).
- `%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 [ACL-Selektoren](#).
- `clearselectors` – löscht alle an den Benutzer angehängten Selektoren.

Anwenden von RBAC auf einen Cache für ElastiCache für Redis

Führen Sie die folgenden Schritte aus, um ElastiCache for Redis RBAC zu verwenden:

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

- [Erstellen von Benutzern und Benutzergruppen mit der Konsole und dem CLI](#)

- [Verwalten von Benutzergruppen mit der Konsole und dem CLI](#)
- [Zuweisen von Benutzergruppen zu Serverless-Caches](#)
- [Zuweisung von Benutzergruppen zu Replikationsgruppen](#)

Erstellen von Benutzern und Benutzergruppen mit der Konsole und dem CLI

Die Benutzerinformationen für RBAC-Benutzer sind eine Benutzer-ID, ein Benutzername, sowie optional ein Kennwort 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 sie allen Benutzergruppen hinzu. Sie können diesen Benutzer nicht löschen oder ändern. Dieser Benutzer ist für die Kompatibilität mit dem Standardverhalten früherer Redis-Versionen gedacht und hat eine Zugriffszeichenfolge, die es ihm erlaubt, 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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-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.

Ändern Sie den Standardbenutzer mit der AWS CLI wie folgt:

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*"
```

2. 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" \  
  --passwords "a-strong-password"
```

```
--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"
```

3. 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 insbesondere diese Benutzerpasswort-Restriktionen, wenn Sie RBAC-für-ElastiCache-for-Redis verwenden:

- Passwörter müssen 16-128 druckbare Zeichen enthalten.
- Folgende nicht-alphanumerische Zeichen sind nicht zulässig: , " " / @.

Verwalten von Benutzern mit der Konsole und dem 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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Amazon-ElastiCache-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.

So ändern Sie einen Benutzer mit dem 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.

So löschen Sie einen Benutzer mit dem 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
```

Verwalten von Benutzergruppen mit der Konsole und dem 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 in der AWS Management Console an und öffnen Sie die Amazon-ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Amazon-ElastiCache-Dashboard 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.

Gehen Sie wie unten beschrieben vor, um die Benutzergruppen im CLI zu verwalten.

So erstellen Sie eine neue Benutzergruppe und fügen einen Benutzer mittels der CLI hinzu

- 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
```

So ändern Sie eine Benutzergruppe durch Hinzufügen neuer Benutzer oder Entfernen aktueller Mitglieder mit der 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.

So löschen Sie eine Benutzergruppe mit dem 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 anzuzeigen, können Sie die [beschreibe-benutzergruppen](#)-Operation verwenden.

```
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 von RBAC darin, diese Benutzergruppe einem Serverless-Cache zuzuweisen.

Zuweisen von Benutzergruppen zu Serverless-Caches mithilfe der Konsole

Zum Hinzufügen einer Benutzergruppe zu einem Serverless-Cache mit der AWS Management Console gehen Sie wie folgt vor:

- Informationen zum deaktivierten Cluster-Modus finden Sie unter [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- Informationen zum aktivierten Cluster-Modus finden Sie unter [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Zuweisen von Benutzergruppen zu Serverless-Caches mithilfe der AWS CLI

Die folgende AWS CLI-Operation erstellt einen Serverless-Cache unter Verwendung des Parameters `user-group-id` mit dem Wert `my-user-group-id`. Ersetzen Sie die Subnetzgruppe `sng-test` durch eine vorhandene Subnetzgruppe.

Hauptparameter

- **--engine** – Der Wert muss `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 Serverless-Cache unter Verwendung des Parameters `user-group-id` 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 [Anzeigen von ElastiCache-Ereignissen](#).

Zuweisung von Benutzergruppen zu Replikationsgruppen

Nachdem Sie eine Benutzergruppe erstellt und Benutzer hinzugefügt haben, besteht der letzte Schritt bei der Implementierung von RBAC darin, diese Benutzergruppe einer Replikationsgruppe zuzuweisen.

Zuweisen von Benutzergruppen zu Replikationsgruppen mit der Konsole

Zum Hinzufügen einer Benutzergruppe zu einer Replikation mit der AWS Management Console gehen Sie wie folgt vor:

- Informationen zum deaktivierten Cluster-Modus finden Sie unter [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#)
- Informationen zum aktivierten Cluster-Modus finden Sie unter [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#)

Zuweisen von Benutzergruppen zu Replikationsgruppen mit dem AWS CLI

Die folgende AWS CLI-Operation 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** – Der Wert muss `redis` sein.
- **--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 \  
  --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"
```

Die folgende AWS CLI-Operation erstellt 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" \  
  --cache-subnet-group "cache-subnet-group"
```

```
--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 [Anzeigen von ElastiCache-Ereignissen](#).

Migrieren von Redis AUTH zu RBAC

Wenn Sie Redis-AUTH wie in [Authentifizieren mit dem Redis-AUTH-Befehl](#) beschrieben verwenden und auf RBAC umstellen möchten, gehen Sie wie folgt vor.

Gehen Sie wie unten beschrieben vor, um von Redis-AUTH über die Konsole zu RBAC zu migrieren.

Migration von RBAC zu Redis-AUTH in der Konsole

1. Melden Sie sich bei AWS Management Console an 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 unten beschrieben vor, um von Redis-AUTH zu RBAC durch Verwendung von CLI zu migrieren.

Migration von RBAC zu Redis-AUTH durch Verwendung von 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 RBAC zu Redis-AUTH

Wenn Sie RBAC verwenden und zu Redis-AUTH migrieren möchten, finden Sie weitere Informationen unter [Migration von RBAC zu Redis-AUTH](#).

Note

Wenn Sie die Zugriffskontrolle in einem ElastiCache-Cache deaktivieren müssen, verwenden Sie dazu die AWS CLI. Weitere Informationen finden Sie unter [the section called “Deaktivieren der Zugriffskontrolle in einem Redis-Cache von ElastiCache”](#).

Automatisches Rotieren von Passwörtern für Benutzer

Mit AWS Secrets Manager können Sie fest codierte Anmeldeinformationen im Code (einschließlich Passwörter) durch einen API-Aufruf an Secrets Manager ersetzen und das Secret programmgesteuert abrufen. 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 Passwörter für ElastiCache für Redis (d. h. geheimen Schlüssel) automatisch mit einer AWS Lambda-Funktion von Secrets Manager rotieren lassen.

Weitere Informationen zu AWS Secrets Manager finden Sie unter [Was ist AWS Secrets Manager?](#)

So verwendet ElastiCache Secrets

Mit Redis 6 führte ElastiCache für Redis [Rollenbasierte Zugriffskontrolle \(RBAC\)](#) ein, um den Redis-Cluster zu sichern. Mit dieser Funktion können bestimmte Verbindungen bezüglich der ausführbaren Befehle und der Tasten, auf die zugegriffen werden kann, eingeschränkt werden. Bei RBAC erstellt der Kunde zwar einen Benutzer mit Passwörtern, die Passwortwerte müssen jedoch manuell im Klartext eingegeben werden und sind für den Bediener sichtbar.

Mit Secrets Manager wird das Passwort von Anwendungen aus Secrets Manager abgerufen, anstatt es manuell einzugeben und in der Anwendungsconfiguration zu speichern. Weitere Informationen hierzu finden Sie unter [So werden ElastiCache-Benutzer dem Secret zugewiesen.](#)

Für die Verwendung von Secrets fallen Kosten an. Preisinformationen finden Sie unter [AWS Secrets Manager – Preise.](#)

So werden ElastiCache-Benutzer dem Secret zugewiesen

Secrets Manager speichert eine Referenz für den zugehörigen Benutzer im `SecretString`-Feld des Secrets. Auf Seiten von ElastiCache gibt es keinen Verweis auf das Secret.

```
{
  "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

Zur Aktivierung der automatischen Passwortrotation von Secrets Manager erstellen Sie eine Lambda-Funktion, die mit der [modify-user](#)-API interagiert, um die Passwörter des Benutzers zu aktualisieren.

Informationen zur Funktionsweise finden Sie unter [Funktionsweise der Drehung.](#)

Note

Für einige AWS-Services empfiehlt AWS, dass Sie sowohl die globalen Bedingungsschlüssel `aws:SourceArn` als auch `aws:SourceAccount` verwenden, um das verwirrte Stellvertreter-Szenario zu vermeiden. Wenn Sie jedoch die `aws:SourceArn`-Bedingung in Ihre Drehungsfunktions-Richtlinie einschließen, kann die Drehungsfunktion nur verwendet werden, um das von diesem ARN angegebene Secret zu rotieren. Es wird empfohlen, nur den Kontextschlüssel `aws:SourceAccount` anzugeben, damit Sie die Drehungsfunktion für mehrere Geheimnisse verwenden können.

Informationen zu eventuell auftretenden Problemen finden Sie unter [Fehlerbehebung bei der AWS-Secrets-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 AWS Management Console an 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 Speichern.
 - f. Wählen Sie Environment variables (Umgebungsvariablen) aus und legen Sie dann Folgendes fest:
 - i. SECRETS_MANAGER_ENDPOINT – <https://secretsmanager.REGION.amazonaws.com>
 - ii. SERET_ARN – der Amazon-Ressourcenname (ARN) des Secrets, das Sie in Schritt 2 erstellt haben.
 - iii. USER_NAME – Benutzername des ElastiCache-Benutzers.
 - iv. Wählen Sie Speichern.
 - g. Wählen Sie Permissions (Berechtigungen) aus.
 - h. Wählen Sie unter Execution role (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 der AWS Management Console – Informationen finden Sie unter [Einrichten einer automatischen Drehung für AWS-Secrets-Manager-Secrets mit der Konsole](#)

Weitere Informationen zum Einrichten eines Rotationszeitplans finden Sie unter [Zeitplanausdrücke in Secrets-Manager-Rotation](#).

- b. Verwenden der AWS CLI – Informationen finden Sie unter [Einrichten einer automatischen Drehung für AWS Secrets Manager-Secrets-Manager-Secrets mit der AWS Command Line Interface](#)

Authentifizieren mit IAM

Themen

- [Übersicht](#)
- [Einschränkungen](#)
- [Aufstellen](#)
- [Herstellen von Verbindungen](#)

Übersicht

Mit der IAM-Authentifizierung können Sie eine Verbindung mit ElastiCache für Redis mithilfe von AWS-IAM-Identitäten authentifizieren, wenn Ihr Cache für die Verwendung der Redis-Version 7 oder höher konfiguriert ist. Auf diese Weise können Sie Ihr Sicherheitsmodell stärken und viele administrative Sicherheitsaufgaben vereinfachen. Mit der IAM-Authentifizierung können Sie außerdem eine detaillierte Zugriffskontrolle für jeden einzelnen ElastiCache-Cache und jeden ElastiCache-Benutzer konfigurieren und den Prinzipien der geringsten Berechtigung folgen. Die IAM-Authentifizierung für ElastiCache für Redis funktioniert, indem ein kurzlebiges IAM-Authentifizierungstoken anstelle eines langlebigen ElastiCache-Benutzerpassworts im Redis-Befehl AUTH oder HELLO bereitgestellt wird. Weitere Informationen zum IAM-Authentifizierungstoken finden Sie im Allgemeinen Referenzhandbuch von AWS beim [Signiervorgang für Signature Version 4](#) und im folgenden Codebeispiel.

Sie können IAM-Identitäten und die zugehörigen Richtlinien verwenden, um den Redis-Zugriff weiter einzuschränken. Sie können Benutzern von ihren Verbundidentitätsanbietern auch direkt Zugriff auf Redis-Caches gewähren.

Um AWS IAM mit ElastiCache für Redis zu verwenden, müssen Sie zuerst einen ElastiCache-Benutzer erstellen, bei dem der Authentifizierungsmodus auf IAM festgelegt ist. Anschließend können Sie eine IAM-Identität erstellen oder wiederverwenden. Die IAM-Identität benötigt eine zugehörige Richtlinie, um die `elasticache:Connect`-Aktion dem ElastiCache-Cache und dem ElastiCache-Benutzer zuzuweisen. Nach der Konfiguration können Sie ein IAM-Authentifizierungstoken mit den AWS-Anmeldeinformationen des IAM-Benutzers oder der IAM-Rolle erstellen. Schließlich müssen Sie das kurzlebige IAM-Authentifizierungstoken als Passwort in Ihrem Redis-Client angeben, wenn Sie eine Verbindung mit Ihrem Redis-Cache herstellen. Ein Redis-Client mit Unterstützung für Anbieter von Anmeldeinformationen kann die temporären Anmeldeinformationen automatisch für jede neue Verbindung generieren. ElastiCache für Redis führt die IAM-Authentifizierung

für Verbindungsanfragen von IAM-fähigen ElastiCache-Benutzern durch und validiert die Verbindungsanfragen mit IAM.

Einschränkungen

Bei der Verwendung der IAM-Authentifizierung gelten die folgenden Einschränkungen:

- Die IAM-Authentifizierung ist verfügbar, wenn Sie ElastiCache für Redis Version 7.0 oder höher verwenden.
- Für IAM-fähige ElastiCache-Benutzer müssen die Eigenschaften für den Benutzernamen und die Benutzer-ID identisch sein.
- Das IAM-Authentifizierungstoken ist für 15 Minuten gültig. Für langlebige Verbindungen empfehlen wir die Verwendung eines Redis-Clients, der eine Schnittstelle zu einem Anbieter von Anmeldeinformationen unterstützt.
- Eine IAM-authentifizierte Verbindung mit ElastiCache für Redis wird nach 12 Stunden automatisch getrennt. Die Verbindung kann um 12 Stunden verlängert werden, indem ein AUTH- oder HELLO-Befehl mit einem neuen IAM-Authentifizierungstoken gesendet wird.
- Die IAM-Authentifizierung wird in MULTI EXEC-Befehlen nicht unterstützt.
- Derzeit unterstützt die IAM-Authentifizierung die folgenden globalen Bedingungskontextschlüssel:
 - Bei Verwendung der IAM-Authentifizierung mit Serverless-Caches werden `aws:VpcSourceIp`, `aws:SourceVpc`, `aws:SourceVpce`, `aws:CurrentTime`, `aws:EpochTime` und `aws:ResourceTag/%s` (von zugehörigen Serverless-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 über globale Bedingungskontextschlüssel finden Sie unter [Globale AWS-Bedingungskontextschlüssel](#) 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" \  
  --tags Key=Value
```



```
--engine redis
```

- Erstellen Sie, wie unten dargestellt, ein Dokument mit den IAM-Vertrauensrichtlinien für Ihre Rolle, sodass Ihr Konto die neue Rolle übernehmen kann. 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 im Folgenden dargestellt. 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
```

6. Fügen 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 [vorab signierten AWS-SigV4-Anfrage](#) generieren. Danach geben Sie, wie im folgenden Beispiel dargestellt, das IAM-Authentifizierungstoken als Passwort an, wenn Sie eine Verbindung mit einem Redis-Cache herstellen.

```
String userId = "insert user id";
```

```
String cacheName = "insert cache name";
boolean isServerless = true;
String region = "insert region";

// Create a default AWS Credentials provider.
// This will look for AWS credentials defined in environment variables or system
// properties.
AWSCredentialsProvider awsCredentialsProvider = new
    DefaultAWSCredentialsProviderChain();

// Create an IAM authentication token request and signed it using the AWS credentials.
// The pre-signed request URL is used as an IAM authentication token for ElastiCache
// Redis.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);
String iamAuthToken =
    iamAuthTokenRequest.toSignedRequestUri(awsCredentialsProvider.getCredentials());

// Construct Redis URL with IAM Auth credentials provider
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(userId, iamAuthToken)
    .build();

// Create a new Lettuce Redis client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

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 mit ElastiCache für Redis mithilfe des Anbieters für Anmeldeinformationen zur IAM-Authentifizierung 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.
IAMAuthTokenRequest iamAuthTokenRequest = new IAMAuthTokenRequest(userId, cacheName,
    region, isServerless);

// Create a Redis credentials provider using IAM credentials.
RedisCredentialsProvider redisCredentialsProvider = new
    RedisIAMAuthCredentialsProvider(
        userId, iamAuthTokenRequest, awsCredentialsProvider);

// Construct Redis URL with IAM Auth credentials provider
```

```
RedisURI redisURI = RedisURI.builder()
    .withHost(host)
    .withPort(port)
    .withSsl(ssl)
    .withAuthentication(redisCredentialsProvider)
    .build();

// Create a new Lettuce Redis client
RedisClient client = RedisClient.create(redisURI);
client.connect();
```

Im Folgenden finden Sie ein Beispiel für einen Lettuce-Redis-Client, der IAMAuthTokenRequest in einen Anbieter von Anmeldeinformationen verpackt, 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());
    }
}
```

```
}
```

Authentifizieren mit dem Redis-AUTH-Befehl

Note

Redis AUTH wurde ersetzt durch. [the section called “Rollenbasierte Zugriffskontrolle \(RBAC\)”](#)
Alle Serverless-Caches müssen RBAC für die Authentifizierung verwenden.

Redis-Authentifizierungstoken oder -Passwörter ermöglichen es Redis, ein Passwort zu verlangen, bevor Clients Befehle ausführen können, wodurch die Datensicherheit verbessert wird. Redis AUTH ist nur für selbst entworfene Cluster verfügbar.

Themen

- [Überblick über AUTH für Redis ElastiCache](#)
- [Authentifizierung auf einen ElastiCache For-Redis-Cluster anwenden](#)
- [Ändern des AUTH-Tokens auf einem vorhandenen Cluster ElastiCache für Redis](#)
- [Migration von RBAC zu Redis-AUTH](#)

Überblick über AUTH für Redis ElastiCache

Wenn Sie Redis AUTH mit Ihrem ElastiCache for Redis-Cluster verwenden, gibt es einige Verbesserungen.

Beachten Sie insbesondere diese Einschränkungen für AUTH-Token oder -Passwörter, wenn Sie AUTH mit for Redis verwenden: ElastiCache

- 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 für Redis-Cluster aktiviert ist. ElastiCache

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.

Authentifizierung auf einen ElastiCache For-Redis-Cluster anwenden

Sie können von den Benutzern verlangen, dass sie ein Token (Passwort) auf einem Token-geschützten Redis-Server eingeben. Geben Sie dazu beim Erstellen Ihrer Replikationsgruppe oder Ihres Clusters den Parameter `--auth-token` (API: `AuthToken`) mit dem richtigen Token an. 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` – Der Wert muss `redis` sein.
- `--engine-version` – Muss 3.2.6, 4.0.10 oder höher sein.
- `--transit-encryption-enabled` – Erforderlich für Authentifizierung und HIPAA-Berechtigung.
- `--auth-token` – Für HIPAA-Konformität erforderlich. Dieser Wert muss das passende Token für diesen Token-geschützten Redis-Server sein.
- `--cache-subnet-group` – Für HIPAA-Konformität erforderlich.

Für Linux, macOS oder Unix:

```
aws elasticache create-replication-group \  
  --replication-group-id authtestgroup \  
  --replication-group-description authtest \  
  --engine redis \  
  --cache-subnet-group sng-test
```



```
--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 ElastiCache für Redis

Um die Aktualisierung Ihrer Authentifizierung zu vereinfachen, können Sie das in einem ElastiCache for AUTH Redis-Cluster verwendete Token ändern. Sie können diese Änderung vornehmen, wenn die Engine-Version 5.0.6 oder höher ist und wenn ElastiCache für Redis die Verschlüsselung bei der Übertragung aktiviert ist.

Das Ändern des Authentifizierungs-Token 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 einziges AUTH-Token unterstützt. Führen Sie diese Änderungsaufrufe mit dem `--apply-immediately`-Parameter durch, um die Änderungen sofort anzuwenden.

Rotation des AUTH-Tokens

Um einen Redis-Server mit einem neuen AUTH-Token zu aktualisieren, rufen Sie die `ModifyReplicationGroup` API mit dem `--auth-token` Parameter als neuem AUTH Token und dem `--auth-token-update-strategy` Wert ROTATE auf. Nach Abschluss der ROTATE-Änderung unterstützt der Cluster zusätzlich zu dem im Parameter angegebenen auch das vorherige AUTH-Token. `auth-token` Wenn vor der AUTH-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 [Setzen des AUTH-Tokens](#) zur Aktualisierung des erforderlichen AUTH-Tokens finden Sie unter Verwendung der Aktualisierungsstrategie SET.

Note

Wenn Sie das AUTH-Token vorher nicht konfigurieren, unterstützt der Cluster nach Abschluss der Änderung zusätzlich zu dem AUTH-Token, das im AUTH-Token-Parameter angegeben ist, kein weiteres AUTH-Token.

Wenn diese Änderung auf einem Server durchgeführt wird, der bereits zwei AUTH-Token unterstützt, wird bei diesem Vorgang auch das älteste AUTH-Token entfernt. Dadurch kann ein Server bis zu zwei aktuelle AUTH-Token gleichzeitig unterstützen.

An dieser Stelle 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 (im folgenden Abschnitt erläutert) verwenden, um ausschließlich den neuen Token zu verwenden.

Mit dem folgenden AWS CLI Vorgang wird eine Replikationsgruppe so geändert, dass das Token rotiert wirdAUTH. *This-is-the-rotated-token*

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
```

Setzen des AUTH-Tokens

Um einen 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 verfügt, wenn zuvor eine ROTATE-Strategie verwendet wurde. Nach Abschluss der Änderung unterstützt der Redis-Server nur das im Parameter `auth-token` angegebene AUTH-Token.

Der folgende AWS CLI Vorgang ändert eine Replikationsgruppe, auf die das AUTH-Token gesetzt 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 ElastiCache für Redis

Um die Authentifizierung auf einem vorhandenen Redis-Server zu aktivieren, rufen Sie die ModifyReplicationGroup-API-Operation auf. Rufen Sie ModifyReplicationGroup mit dem `--auth-token`-Parameter als neuem Token und `--auth-token-update-strategy` mit dem Wert `ROTATE` auf.

Nach Abschluss der ROTATE-Änderung unterstützt der Cluster das im `--auth-token` Parameter angegebene AUTH Token zusätzlich zur Unterstützung von Verbindungen ohne Authentifizierung. Sobald alle Client-Anwendungen für die Authentifizierung bei Redis mit dem AUTH-Token aktualisiert wurden, verwenden Sie die SET-Strategie, um das AUTH-Token als erforderlich zu kennzeichnen.

Die Aktivierung der Authentifizierung wird nur auf Redis-Servern mit aktivierter Verschlüsselung bei der Übertragung (TLS) unterstützt.

Migration von RBAC zu Redis-AUTH

Wenn Sie Benutzer mit Redis Role-Based Access Control (RBAC) wie unter beschrieben authentifizieren und zu Redis AUTH migrieren möchten [Rollenbasierte Zugriffskontrolle \(RBAC\)](#), verwenden Sie die folgenden Verfahren. Sie können entweder mit der Konsole oder mit dem CLI migrieren.

Migration von RBAC zu Redis-AUTH über die Konsole

1. [Melden Sie sich bei der an und öffnen Sie die Konsole unter https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/). [AWS Management Console 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 Zugriffskontrolle Sie die Option Redis AUTH-Standardbenutzerzugriff aus.
7. Legen Sie unter Redis AUTH-Token ein neues Token fest.
8. Wählen Sie Änderungen in der Vorschau anzeigen und dann auf dem nächsten Bildschirm Ändern aus.

Um von RBAC zu Redis AUTH zu migrieren, verwenden Sie den AWS CLI

Verwenden Sie einen der folgenden Befehle, um ein neues optionales AUTH Token für Ihre 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 markiert ist. Verwenden Sie dabei die Aktualisierungsstrategie SET im folgenden Schritt.

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 Redis-Anwendungen so aktualisieren, dass sie sich mit dem neu konfigurierten optionalen AUTH-Token bei der ElastiCache Replikationsgruppe authentifizieren. 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 Redis-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 [Authentifizieren mit dem Redis-AUTH-Befehl](#).

 Note

Informationen zum Deaktivieren der Zugriffskontrolle auf einem ElastiCache Cluster finden Sie unter [the section called “Deaktivieren der Zugriffskontrolle in einem Redis-Cache von ElastiCache”](#)

Deaktivieren der Zugriffskontrolle in einem Redis-Cache von ElastiCache

Folgen Sie den nachstehenden Anweisungen, um die Zugriffskontrolle in einem TLS-fähigen Redis-Cache zu deaktivieren. Für Ihren Redis-Cache sind zwei Arten von Konfigurationen möglich: Redis AUTH-Standardbenutzerzugriff oder User Group Access Control List (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.

So deaktivieren Sie einen Redis-Serverless-Cache, der mit RBAC konfiguriert ist

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": ""
  "...",
}
```

So deaktivieren Sie einen Redis-Cache, der mit einem AUTH-Token konfiguriert ist

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
```

So deaktivieren Sie einen Redis-Cluster, der mit RBAC konfiguriert ist

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 Methoden, um die 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.

Amazon VPCs und ElastiCache Sicherheit

Da die Sicherheit der Daten wichtig ist, stellt ElastiCache Methoden zur Verfügung, mit denen Sie kontrollieren können, 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 (Amazon VPC) oder Amazon EC2-Classic gestartet haben oder ob nicht.

 **Important**

Die Verwendung von Amazon EC2-Classic zum Starten von ElastiCache-Clustern wird nicht mehr unterstützt. Alle aktuellen Generierungsknoten werden ausschließlich in Amazon Virtual Private Cloud gestartet.

Der Amazon Virtual Private Cloud (Amazon VPC)-Service definiert ein virtuelles Netzwerk, das einem herkömmlichen Rechenzentrum sehr ähnlich ist. Wenn Sie Ihre Amazon VPC konfigurieren, können Sie den IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen konfigurieren. Sie können dem virtuellen Netzwerk außerdem einen Cache-Cluster hinzufügen und den Zugriff auf den Cache-Cluster über Amazon-VPC-Sicherheitsgruppen steuern.

In diesem Abschnitt wird erklärt, wie ein ElastiCache-Cluster manuell in einer Amazon VPC konfiguriert wird. Diese Informationen richten sich an Benutzer, die mehr darüber wissen möchten, wie ElastiCache und Amazon VPC miteinander funktionieren.

Themen

- [Grundlegendes zu ElastiCache und Amazon VPCs](#)
- [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#)
- [Erstellen einer Virtual Private Cloud \(VPC\)](#)
- [Herstellen einer Verbindung mit einem Cache, der in einer Amazon VPC ausgeführt wird](#)

Grundlegendes zu ElastiCache und Amazon VPCs

ElastiCache ist vollständig in die Amazon Virtual Private Cloud (Amazon VPC) integriert. Für ElastiCache-Benutzer bedeutet dies Folgendes:

- Wenn Ihr AWS-Konto nur die EC2-VPC-Plattform unterstützt, startet ElastiCache Ihren Cluster immer in einer Amazon VPC.
- Wenn Sie neu bei AWS sind, werden Ihre Cluster in einer Amazon VPC bereitgestellt. Eine Standard-VPC wird automatisch für Sie erstellt.
- Wenn Sie eine Standard-VPC haben und beim Starten eines Clusters kein Subnetz angeben, wird der Cluster in Ihrer Standard-Amazon-VPC gestartet.

Weitere Informationen finden Sie unter [Detecting Your Supported Platforms and Whether You Have a Default VPC](#).

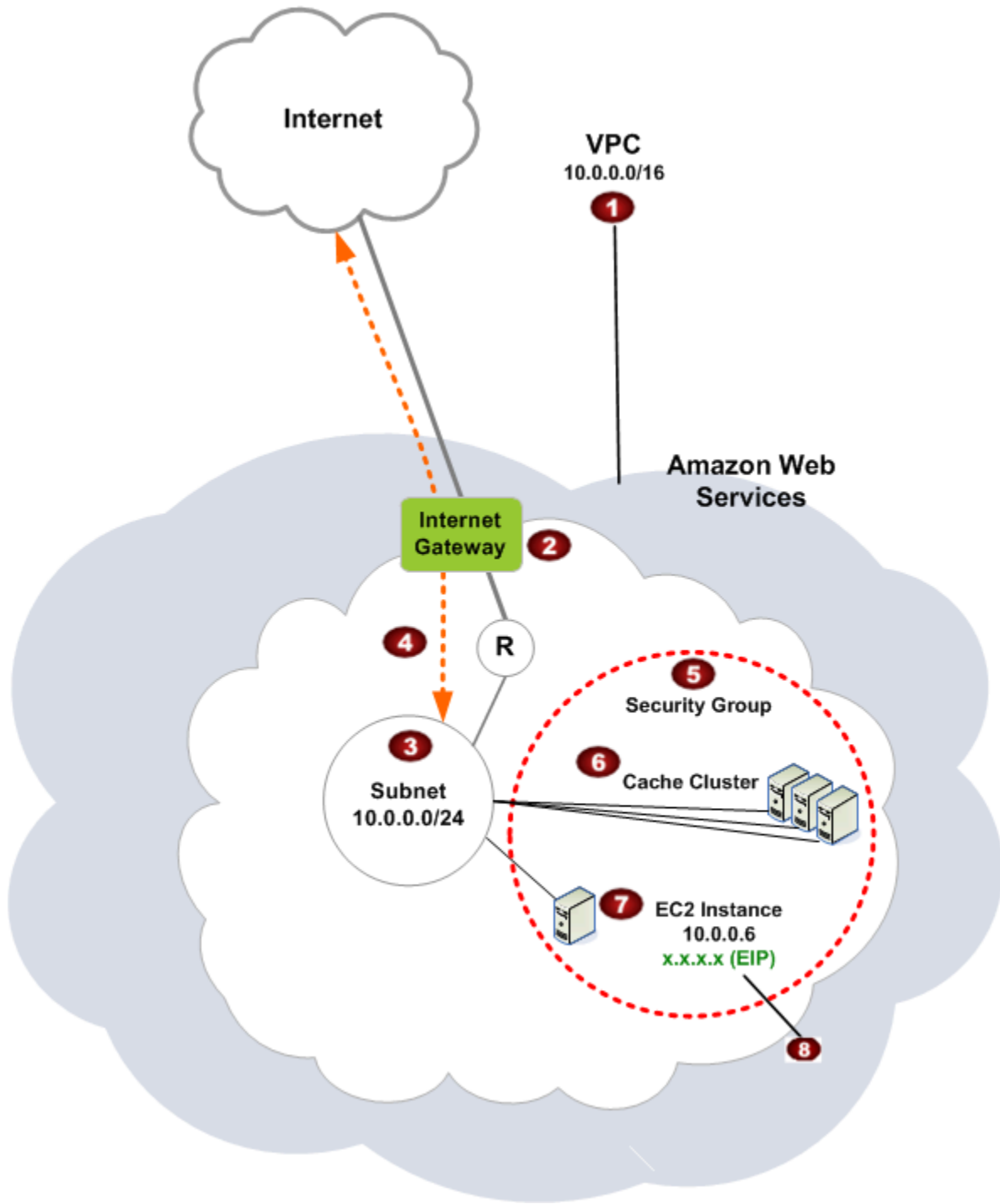
Mit AWS können Sie ein virtuelles Netzwerk in der Amazon Virtual Private Cloud erstellen, das einem herkömmlichen Rechenzentrum sehr ähnelt. Sie können Ihre Amazon VPC konfigurieren und den IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen konfigurieren.

Die Grundfunktionalität von ElastiCache bleibt in einer Virtual Private Cloud dieselbe. ElastiCache verwaltet Software-Upgrades, Patch-Vorgänge, Fehlererkennung und Wiederherstellung, unabhängig davon, ob Ihre Cluster innerhalb oder außerhalb einer Amazon VPC bereitgestellt werden.

ElastiCache-Cache-Knoten, die außerhalb einer Amazon VPC bereitgestellt werden, werden einer IP-Adresse zugeordnet, zu der der Endpunkt/DNS-Name aufgelöst wird. Dies bietet Konnektivität von Amazon Elastic Compute Cloud (Amazon EC2) -Instances. Wenn Sie einen ElastiCache-Cluster in einem privaten Amazon-VPC-Subnetz starten, wird jeder Cache-Knoten einer privaten IP-Adresse innerhalb dieses Subnetzes zugeordnet.

Überblick über ElastiCache in einer Amazon VPC

In der folgenden Tabelle werden die Amazon VPC Umgebung sowie ElastiCache Cluster und Amazon EC2 Instances beschrieben, die in der Amazon VPC gestartet werden.



1

Die Amazon VPC ist ein isolierter Bereich der AWS-Cloud, dem ein eigener Block von IP-Adressen zugeordnet ist.

2

Ein Internet-Gateway verbindet Ihre Amazon-VPC direkt mit dem Internet und bietet Zugriff auf andere AWS-Ressourcen wie Amazon Simple Storage Service (Amazon S3), die außerhalb Ihrer Amazon VPC ausgeführt werden.

3

Ein Amazon-VPC-Subnetz ist ein Segment des IP-Adressbereichs einer Amazon VPC, indem Sie AWS-Ressourcen entsprechend Ihren Sicherheits- und Betriebsanforderungen isolieren können.

4

Eine Routing-Tabelle in der Amazon VPC steuert den Netzwerkdatenverkehr zwischen dem Subnetz und dem Internet. Die Amazon VPC hat einen impliziten Router, der in diesem Diagramm durch den Kreis mit dem R dargestellt wird.

5

Eine Amazon VPC -Sicherheitsgruppe steuert den eingehenden und ausgehenden Datenverkehr für Ihre ElastiCache Cluster und Amazon EC2 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 Amazon-EC2-Instances auch im Subnetz starten. Jede Amazon-EC2-Instanz hat eine private IP-Adresse aus dem Adressbereich des Subnetzes. Die Amazon-EC2-Instanz kann eine Verbindung zu jedem Cache-Knoten in demselben Subnetz herstellen.

8

Damit eine Amazon-EC2-Instanz in Ihrer Amazon VPC vom Internet aus erreichbar ist, müssen Sie der Instanz eine statische, öffentliche Adresse namens Elastic IP-Adresse zuordnen.

Voraussetzungen

Zum Erstellen eines ElastiCache Clusters innerhalb einer Amazon VPC muss Ihre Amazon VPC die folgenden Anforderungen erfüllen:

- Die Amazon VPC muss nicht dedizierte Amazon EC2 Instances zulassen. Sie können ElastiCache nicht in einer Amazon VPC verwenden, die für eine Dedicated Instance-Tenancy konfiguriert ist.

- Für Ihre Amazon VPC muss eine Cache-Subnetzgruppe definiert werden. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen und diese mit Ihren VPC-Endpunkten oder Cache-Knoten zu verknüpfen.
- CIDR-Blöcke für jedes Subnetz müssen groß genug sein, um freie IP-Adressen für ElastiCache bereitzustellen, die während Wartungsarbeiten genutzt werden können.

Weiterleitung und Sicherheit

Sie können Weiterleitung in Ihrer Amazon VPC so konfigurieren, dass gesteuert wird, wo der Datenverkehr fließt (z. B. zum Internet-Gateway oder Virtual Private Gateway). Mit einem Internet-Gateway hat Ihre Amazon VPC direkten Zugriff auf andere AWS-Ressourcen, die nicht in Ihrer Amazon VPC ausgeführt werden. Wenn Sie sich nur für ein virtuelles privates Gateway mit einer Verbindung zum lokalen Netzwerk Ihrer Organisation entscheiden, können Sie Ihren Internet-Datenverkehr über das VPN leiten und lokale Sicherheitsrichtlinien und eine Firewall verwenden, um ausgehenden Datenverkehr zu steuern. In diesem Fall können zusätzliche Bandbreitengebühren anfallen, wenn Sie über das Internet auf AWS-Ressourcen zugreifen.

Sie können Amazon VPC-Sicherheitsgruppen verwenden, um die ElastiCache Cluster und Amazon EC2 Instances in Ihrer Amazon VPC zu schützen. Sicherheitsgruppen wirken wie eine Firewall auf der Instance-Ebene, nicht auf der Subnetzebene.

Note

Es wird jedoch dringend empfohlen, DNS-Namen für die Verbindung mit Ihren Cache-Knoten zu verwenden, da sich die zugrunde liegende IP-Adresse ändern kann.

Amazon VPC-Dokumentation

Amazon VPC hat eine eigene Dokumentation, in der das Erstellen und Nutzen Ihrer erklärt wird. In der folgenden Tabelle sind Links zu den Amazon-VPC-Handbüchern aufgeführt.

| Beschreibung | Dokumentation |
|--|---|
| Erste Schritte bei der Verwendung von Amazon VPC | Erste Schritte mit Amazon VPC |

| Beschreibung | Dokumentation |
|---|---|
| Verwenden Sie Amazon VPC über die AWS Management Console | Amazon VPC User Guide |
| Vollständige Beschreibungen aller Amazon-VP C-Befehle | Amazon EC2 Command Line Reference (Die Amazon VPC Befehle befinden sich in der Amazon EC2 Referenz) |
| Vollständige Beschreibungen der Amazon-VP C-API-Aktionen, -Datentypen und -Fehler | Amazon-EC2-API-Referenz (Die Amazon VPC API-Vorgänge befinden sich in der Amazon EC2 Referenz) |
| Informationen für den Netzwerkadministrator, der das Gateway an Ihrem Ende einer optionalen IPsec-VPN-Verbindung konfigurieren muss | Was ist AWS-Site-to-Site-VPN? |

Weitere Informationen zur Amazon Virtual Private Cloud finden Sie unter [Amazon Virtual Private Cloud](#).

Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC

Amazon ElastiCache unterstützt die folgenden Szenarien für den Zugriff auf einen Cache in einer Amazon VPC:

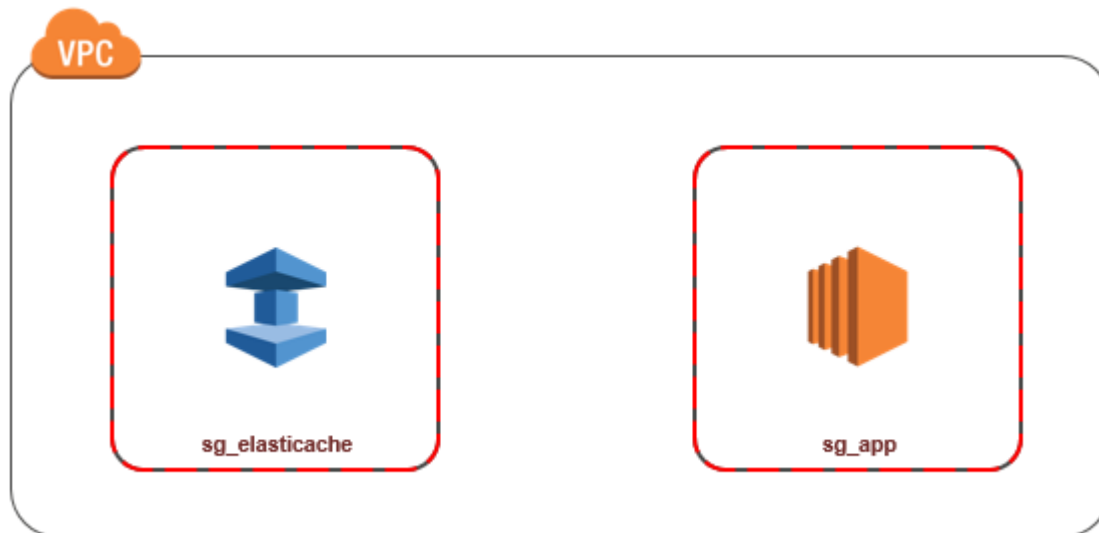
Inhalt

- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in derselben Amazon VPC befinden](#)
- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in unterschiedlichen Amazon VPCs befinden](#)
 - [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in verschiedenen Amazon VPCs in derselben Region befinden](#)
 - [Verwenden von Transit Gateway](#)
 - [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in unterschiedlichen Amazon VPCs in verschiedenen Regionen befinden](#)
 - [Verwenden von Transit VPC](#)
- [Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden ausgeführt wird](#)
 - [Zugriff 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 Amazon EC2 EC2-Instance in derselben Amazon VPC befinden

Der gängigste Anwendungsfall ist, wenn eine auf einer EC2-Instance bereitgestellte Anwendung eine Verbindung mit einem Cache in der gleichen VPC herstellen muss.

Das folgende Diagramm veranschaulicht dieses Szenario.



Nachfolgend finden Sie den einfachsten Weg für die Verwaltung des Zugriffs zwischen EC2-Instances und Caches in derselben VPC:

1. Erstellen Sie eine Sicherheitsgruppe für Ihren Cache. Diese Sicherheitsgruppe kann verwendet werden, um den Zugriff auf den Cache zu beschränken. Sie können für diese Sicherheitsgruppe beispielsweise eine benutzerdefinierte Regel erstellen, die TCP-Zugriff über den Port, den Sie dem Cache bei seiner Erstellung zugeordnet haben, und eine IP-Adresse gewährt, mit der Sie auf den Cache zugreifen.

Der Standardport für Redis-Caches ist 6379.

2. Erstellen Sie eine VPC-Sicherheitsgruppe für Ihre EC2-Instances (Web- und Anwendungsserver). Mithilfe dieser Sicherheitsgruppe können Sie bei Bedarf den Internetzugriff auf die EC2-Instance über die Routing-Tabelle der VPC zulassen. Sie können beispielsweise Regeln für diese Sicherheitsgruppe festlegen, damit der TCP-Zugriff auf die EC2-Instance über Port 22 möglich ist.
3. Erstellen Sie benutzerdefinierte Regeln in der Sicherheitsgruppe für Ihren Cache, die Verbindungen über die von Ihnen für EC2-Instances erstellte Sicherheitsgruppe zulassen. 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 Local Zone erstellen, wird Ihre VPC auf diese Local Zone erweitert, und Ihre VPC behandelt das Subnetz wie jedes Subnetz in einer anderen

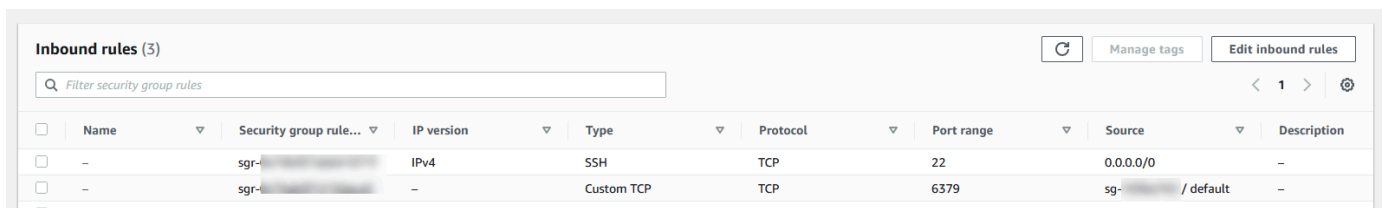
Availability Zone. Alle relevanten Gateways und Routing-Tabellen werden automatisch angepasst.

So erstellen Sie eine Regel in einer VPC-Sicherheitsgruppe, die Verbindungen über eine andere Sicherheitsgruppe zulässt

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon VPC-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 für Type die Option Custom TCP Rule aus.
 - a. Geben Sie für Port-Bereich den Port an, den Sie beim Erstellen des Caches verwendet haben.

Der Standard-Port für 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 Amazon-EC2-Instances verwenden werden.
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 Amazon EC2 EC2-Instance in unterschiedlichen Amazon VPCs befinden

Wenn sich Ihr Cache in einer anderen VPC als die EC2-Instance befindet, mit der Sie auf ihn zugreifen, kann der Zugriff auf den Cache auf verschiedene Arten erfolgen. Wenn sich der Cache und die EC2-Instance in verschiedenen VPCs, aber in derselben Region befinden, können Sie

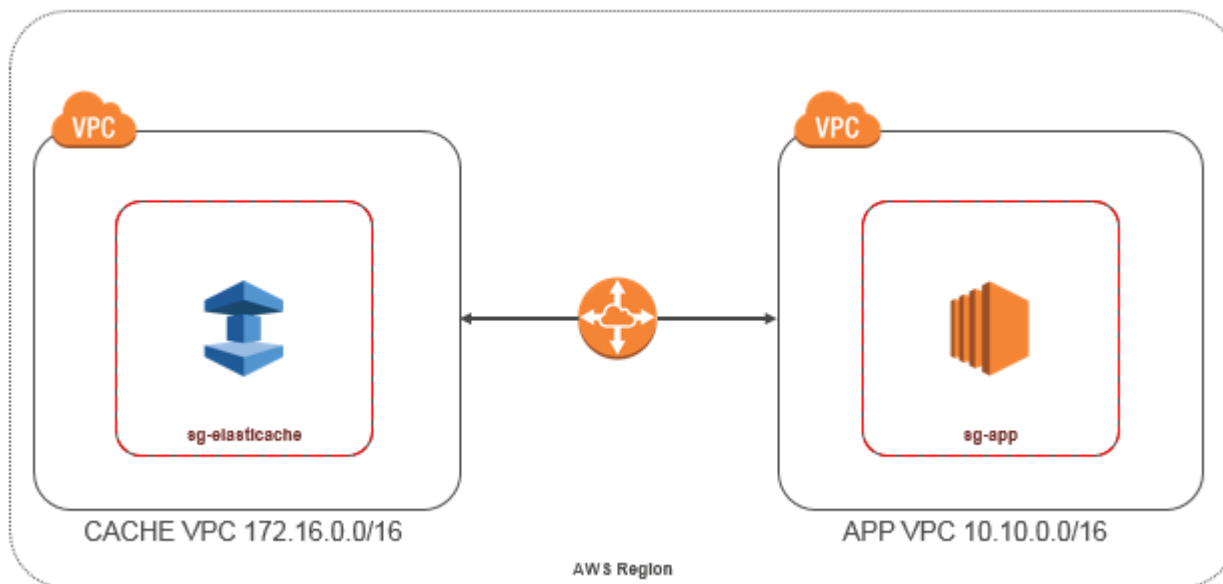
VPC-Peering verwenden. Wenn sich der Cache und die EC2-Instance in verschiedenen Regionen befinden, können Sie VPN-Konnektivität zwischen den Regionen aufbauen.

Themen

- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in verschiedenen Amazon VPCs in derselben Region befinden](#)
- [Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in unterschiedlichen Amazon VPCs in verschiedenen Regionen befinden](#)

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in verschiedenen Amazon VPCs in derselben Region befinden

Das folgende Diagramm veranschaulicht den Zugriff auf einen Cache über eine Amazon-EC2-Instance in einer anderen Amazon VPC in derselben Region mittels einer VPC-Peering-Verbindung.



Cache-Zugriff über eine Amazon-EC2-Instance in einer anderen Amazon VPC innerhalb derselben Region – VPC-Peering-Verbindung

Peering: Eine VPC-Peering-Verbindung ist eine Netzwerkverbindung zwischen zwei VPCs. Diese ermöglicht die Weiterleitung des Datenverkehrs zwischen den VPCs mithilfe von privaten IP-Adressen. Instances in jeder der VPCs können so miteinander kommunizieren, als befänden sie sich im selben Netzwerk. Sie können eine VPC-Peering-Verbindung zwischen Ihren eigenen Amazon

VPCs oder mit einer Amazon VPC in einem anderen AWS Konto innerhalb einer einzelnen Region herstellen. Weitere Informationen zum Amazon-VPC-Peering finden Sie in der [VPC-Dokumentation](#)

Note

Je nach den auf die VPC angewendeten Konfigurationen kann die DNS-Namensauflösung für gepeerte VPCs fehlschlagen. ElastiCache Beide VPCs müssen für DNS-Hostnamen und DNS-Auflösung aktiviert sein, um dies aufzulösen. Weitere Informationen finden Sie unter [Aktivieren einer DNS-Auflösung für eine VPC-Peering-Verbindung](#).

So greifen Sie auf einen Cache in einer anderen Amazon VPC über Peering zu

1. Vergewissern Sie sich, dass sich der IP-Bereich der beiden VPCs nicht überlappt. Andernfalls ist kein Peering zwischen ihnen möglich.
2. Stellen Sie eine Peering-Verbindung zwischen den beiden VPCs her. Weitere Informationen finden Sie unter [Erstellen und Akzeptieren einer Amazon-VPC-Peering-Verbindung](#).
3. Aktualisieren Sie Ihre Routing-Tabelle. Weitere Informationen finden Sie unter [Aktualisieren der 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, um eingehende Verbindungen von der Anwendungssicherheitsgruppe in der Peering-VPC zuzulassen. Weitere Informationen finden Sie unter [Verweisen auf Peer-VPC-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 und VPN-Verbindungen in derselben AWS Region anzuhängen und den Datenverkehr 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, kann der Kontoinhaber seine VPCs an Ihr Transit-Gateway anhängen. Benutzer in einem der Konten können die Anhang jederzeit löschen.

Sie können Multicast auf einem Transit Gateway aktivieren und dann eine Transit Gateway-Multicast-Domain erstellen, mit der Multicast-Datenverkehr von der Multicast-Quelle über VPC-Anhängen, die Sie der Domain zuordnen, an Multicast-Gruppenmitglieder gesendet werden kann.

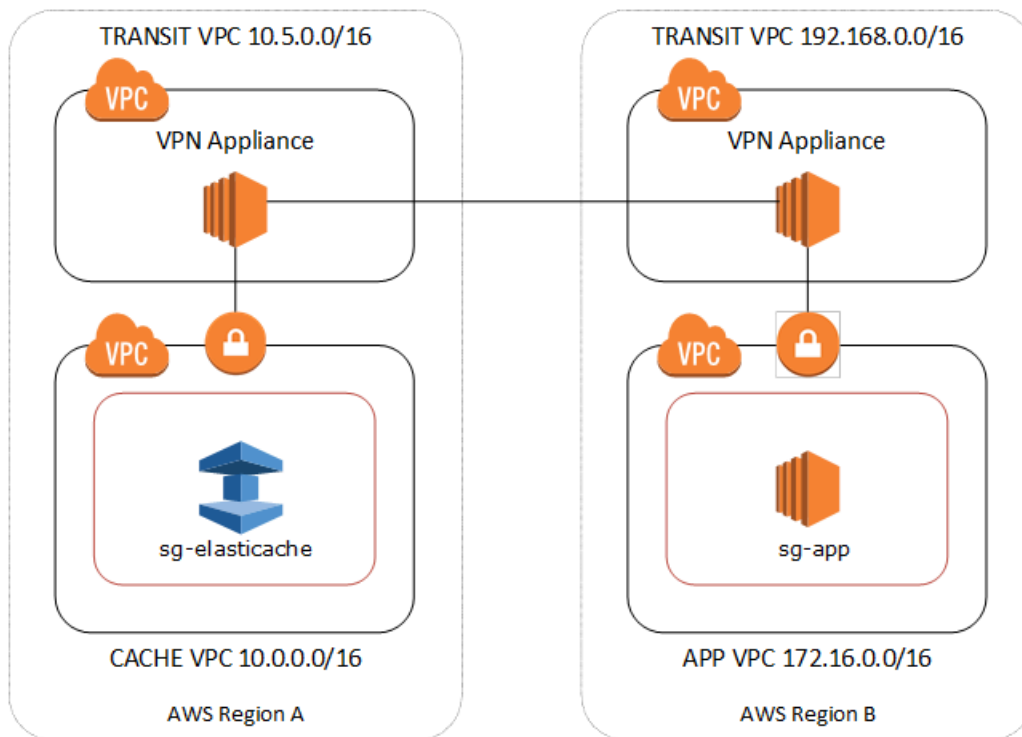
Sie können auch einen Peering-Verbindungsanhang zwischen Transit-Gateways in verschiedenen Regionen erstellen. AWS Auf diese Weise können Sie den Datenverkehr zwischen den Anhängen der Transit Gateways über verschiedene Regionen hinweg leiten.

Weitere Informationen finden Sie unter [Transit Gateways](#).

Zugreifen auf einen ElastiCache Cache, wenn sich dieser und die Amazon EC2 EC2-Instance in unterschiedlichen Amazon VPCs in verschiedenen Regionen befinden

Verwenden von Transit VPC

Eine gängige Strategie zum Herstellen einer Verbindung zwischen vielen, geografisch verstreuten VPCs und Remote-Netzwerken besteht darin, eine Transit-VPC zu erstellen, die als globales Netzwerk-Transit-Zentrum fungiert. Eine Transit-VPC vereinfacht die Netzwerkverwaltung und minimiert die Anzahl von Verbindungen, die für die Vernetzung mehrerer VPCs mit Remote-Netzwerken 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.



Herstellen einer Verbindung zwischen verschiedenen VPCs in verschiedenen Regionen

Sobald die Transit Amazon VPC eingerichtet ist, kann eine Anwendung, die in einer „Spoke“ VPC in einer Region bereitgestellt wird, eine Verbindung zu einem ElastiCache Cache in einer „Spoke“ VPC in einer anderen Region herstellen.

So greifen Sie auf einen Cache in einer anderen VPC in einer anderen AWS Region zu

1. Stellen Sie eine Transit-VPC-Lösung bereit. Weitere Informationen finden Sie unter [AWS - Transit-Gateway](#).
2. Aktualisieren Sie die VPC-Routing-Tabellen in den App- und Cache-VPCs, um den Datenverkehr durch das VGW (Virtual Private Gateway) und die VPN Appliance weiterzuleiten. Im Falle des dynamischen Routing mit Border Gateway Protocol (BGP) werden Ihre Routen möglicherweise automatisch gefüllt.
3. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen aus dem IP-Bereich der Anwendungsinstanzen 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

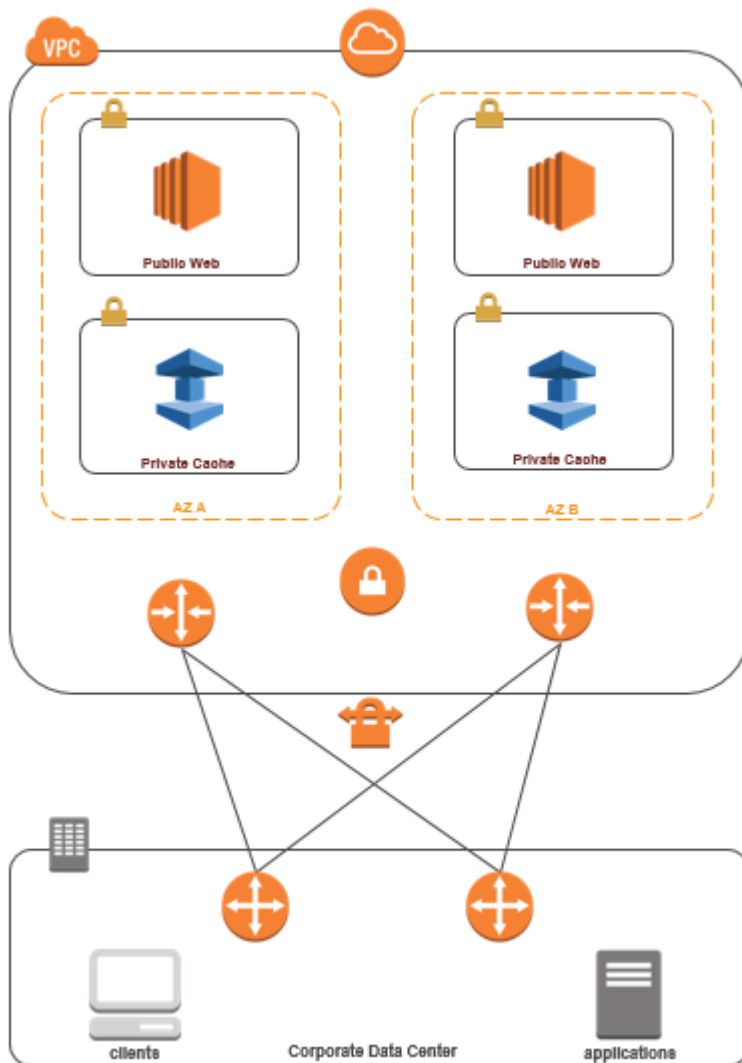
Ein anderes mögliches Szenario ist eine Hybridarchitektur, bei der Clients oder Anwendungen im Rechenzentrum des Kunden möglicherweise auf einen ElastiCache Cache in der VPC zugreifen müssen. Dieses Szenario wird auch unterstützt, vorausgesetzt, dass entweder über VPN oder Direct Connect Konnektivität zwischen der VPC des Kunden und dem Rechenzentrum besteht.

Themen

- [Zugriff 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](#)

Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die im Rechenzentrum eines Kunden mithilfe von VPN-Konnektivität ausgeführt wird

Das folgende Diagramm zeigt den Zugriff auf einen ElastiCache Cache von einer Anwendung aus, die in Ihrem Unternehmensnetzwerk ausgeführt wird, mithilfe von VPN-Verbindungen.



Über ein VPN wird eine Verbindung zu ElastiCache Ihrem Rechenzentrum hergestellt

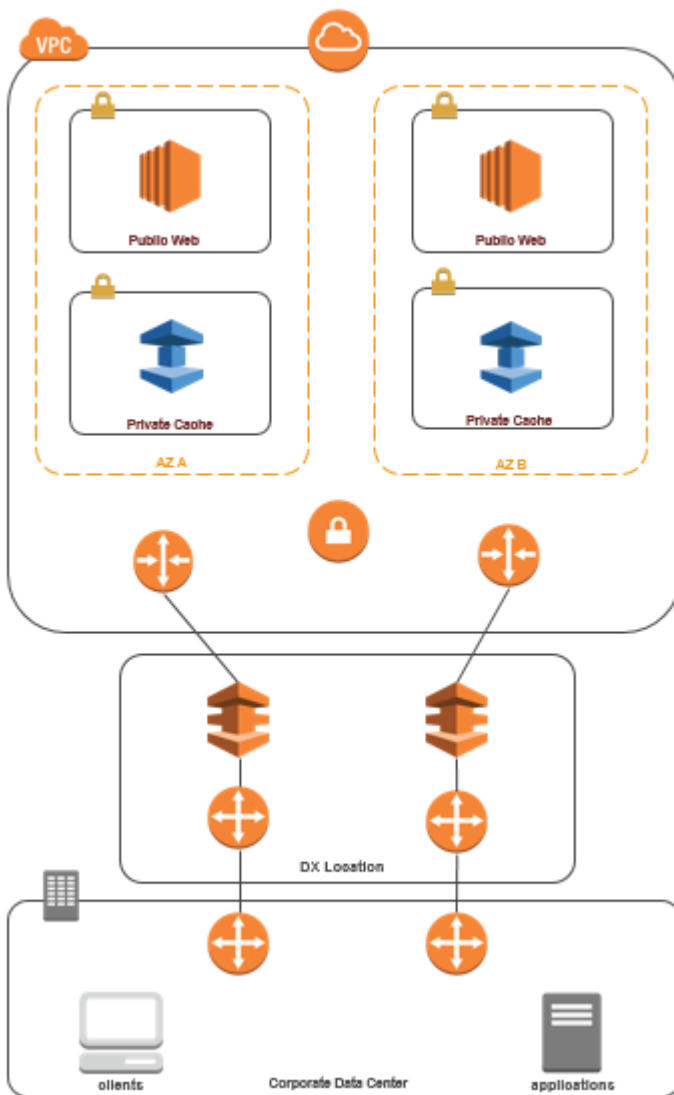
So greifen Sie auf einen Cache in einer VPC von einer Anwendung vor Ort über eine VPN-Verbindung zu

1. Richten Sie VPN-Konnektivität ein, indem Sie ein Hardware Virtual Private Gateway zu Ihrer VPC hinzufügen. Weitere Informationen finden Sie unter [Hinzufügen eines Hardware Virtual Private Gateway zu Ihrer 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 Falle des dynamischen Routing mit BGP werden Ihre Routen möglicherweise automatisch gefüllt.
3. Ändern Sie die Sicherheitsgruppe Ihres ElastiCache Caches, um eingehende Verbindungen von den lokalen Anwendungsservern zuzulassen.

Beim Zugriff auf einen Cache über eine VPN-Verbindung entstehen Netzwerklatenzen und fallen zusätzliche Datenübertragungskosten an.

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.

Erstellen einer Virtual Private Cloud (VPC)

In diesem Beispiel erstellen Sie eine Amazon VPC mit einem privaten Subnetz für jede Availability Zone.

Erstellen einer Amazon VPC (Konsole)

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie auf dem VPC-Dashboard Create VPC (VPC erstellen) aus.
3. Wählen Sie unter Resources to create (Zu erstellende Ressourcen) die Option VPC and more (VPC und mehr) aus.
4. Wählen Sie unter Number of Availability Zones (AZs) (Anzahl der Availability Zones (AZs)) die Anzahl der Availability Zones aus, in denen Sie Ihre Subnetze starten möchten.
5. Wählen Sie unter Number of public subnets (Anzahl der öffentlichen Subnetze) die Anzahl der öffentlichen Subnetze aus, die Sie zu Ihrer VPC hinzufügen möchten.
6. Wählen Sie unter Number of private subnets (Anzahl der privaten Subnetze) die Anzahl der privaten Subnetze aus, die Sie zu Ihrer VPC hinzufügen möchten.

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 Ihrer Amazon VPC eine Amazon-EC2-Instance hinzufügen.

7. Erstellen Sie eine Amazon-VPC-Sicherheitsgruppe. Sie werden diese Gruppe für Ihren Cache-Cluster und Ihre Amazon-EC2-Instance verwenden.
 - a. Wählen Sie im Navigationsbereich der Amazon-VPC-Managementkonsole Sicherheitsgruppen.
 - b. Wählen Sie Sicherheitsgruppen erstellen aus.
 - c. Geben Sie in den entsprechenden Feldern einen Namen und eine Beschreibung für Ihre Sicherheitsgruppe ein. Wählen Sie im Feld VPC die ID für Ihre Amazon VPC aus.

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name [Info](#)

Name cannot be edited after creation.

Description [Info](#)

VPC [Info](#)

Inbound rules [Info](#)

This security group has no inbound rules.

Outbound rules [Info](#)

| Type | Protocol | Port range | Destination | Description - optional |
|-------------|----------|------------|-------------|------------------------|
| All traffic | All | All | Custom | |

- d. Wenn Sie die gewünschten Einstellungen vorgenommen haben, wählen Sie Ja, erstellen aus.
8. Definieren Sie eine Netzwerkeingangsregel für Ihre Sicherheitsgruppe. Diese Regel wird Ihnen ermöglichen, mithilfe von Secure Shell (SSH) eine Verbindung zu Ihrer Amazon-EC2-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 Create a new rule die Option SSH und anschließend Add Rule aus.
 - d. Stellen Sie die folgenden Werte für Ihre neue eingehende Regel ein, um HTTP-Zugriff zuzulassen:
 - Typ: HTTP
 - Quelle: 0.0.0.0/0

Wählen Sie Apply Rule Changes aus.

Sie können nun eine Cache-Subnetzgruppe erstellen und einen Cache-Cluster in Ihrer Amazon VPC starten.

- [Erstellen einer Subnetzgruppe](#)

- [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Herstellen einer Verbindung mit einem Cache, der in einer Amazon VPC ausgeführt wird

In diesem Beispiel wird gezeigt, wie Sie eine Amazon EC2 Instance in Ihrer Amazon VPC starten. Sie können sich dann bei dieser Instance anmelden und auf den ElastiCache-Cache zugreifen, der in der Amazon VPC ausgeführt wird.

Herstellen einer Verbindung mit einem Cache, der in einer Amazon VPC ausgeführt wird (Konsole)

In diesem Beispiel erstellen Sie eine Amazon EC2 Instance in Ihrer Amazon VPC. Sie können diese Amazon EC2 Instance verwenden, um eine Verbindung zu Cache-Knoten herzustellen, die in der Amazon VPC ausgeführt werden.

Note

Weitere Informationen zur Verwendung von Amazon EC2 finden Sie im [Handbuch „Erste Schritte“ zu Amazon EC2](#) in der [Amazon-EC2-Dokumentation](#).

So erstellen Sie eine Amazon EC2 Instance in Ihrer Amazon VPC mithilfe der Amazon EC2 Konsole

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-EC2-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) (Amazon Machine Image (AMI) auswählen) das 64-Bit-Amazon Linux-AMI aus und klicken Sie auf Select (Auswählen).
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 der Liste Netzwerk Ihre Amazon VPC aus.
 - 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 Amazon-EC2-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 [Amazon EC2-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 Amazon-VPC-Sicherheitsgruppe aus und wählen Sie anschließend Prüfen 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 über das Verwalten von Schlüsselpaaren finden Sie im [Handbuch Erste Schritte zu Amazon EC2](#).

11. Wenn Sie bereit sind, Ihre Amazon-EC2-Instance zu starten, wählen Sie Starten aus.

Sie können der Amazon EC2 Instance nun eine Elastic IP-Adresse zuweisen, die Sie gerade erstellt haben. Sie müssen diese IP-Adresse verwenden, um eine Verbindung zur Amazon EC2 Instance herzustellen.

So weisen Sie eine Elastic IP-Adresse zu (Konsole)

1. Öffnen Sie die Amazon-VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie in der Navigationsliste die Option Elastic IPs aus.
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 Dialogfenster Associate Address (Adresse zuweisen) im Feld Instance die ID der Amazon-EC2-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 nun SSH verwenden, um eine Verbindung mit der Amazon EC2 Instance über die von Ihnen erstellte Elastic IP-Adresse herzustellen.

Stellen Sie eine Verbindung zu Ihrer Amazon-EC2-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 Amazon EC2 Instance ab.

Sie können nun mit Ihrem ElastiCache Cluster 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!
```


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 auf <https://console.aws.amazon.com/elasticache/> Um den Endpunkt für einen der Knoten in Ihrem Cache-Cluster abzurufen. Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#) für Redis.
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 Kosten auf Ihrem AWS-Konto zu vermeiden, stellen Sie sicher, dass Sie nach dem Testen dieser Beispiele alle AWS-Ressourcen löschen, die Sie nicht mehr benötigen.

Amazon-ElastiCache-API und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)

Sie können eine private Verbindung zwischen Ihrer VPC und Amazon-ElastiCache-API-Endpunkten herstellen, indem Sie einen Schnittstellen-VPC-Endpunkt erstellen. Schnittstellenendpunkte werden von [AWS PrivateLink](#) betrieben. AWS PrivateLink erlaubt Ihnen den privaten Zugriff auf Amazon-ElastiCache-API-Operationen ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS-Direct-Connect-Verbindung.

Die Instances in Ihrer VPC benötigen für die Kommunikation mit dem Amazon-ElastiCache-API-Endpunkt keine öffentlichen IP-Adressen. Ihre Instances benötigen auch keine öffentlichen IP-Adressen, um beliebige der verfügbaren ElastiCache-API-Operationen zu verwenden. Der Datenverkehr zwischen der VPC und Amazon ElastiCache verlässt das Amazon-Netzwerk nicht. Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic Network-Schnittstellen in Ihren Subnetzen dargestellt. Weitere Informationen zu Elastic Network-Schnittstellen finden Sie unter [Elastic Network-Schnittstellen](#) im Amazon EC2 Benutzerhandbuch.

- Weitere Informationen zu VPC-Endpunkten finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) im Amazon VPC Benutzerhandbuch.
- Weitere Informationen zum Aufrufen von ElastiCache-API-Operationen finden Sie unter [ElastiCache-API-Operationen](#).

Wenn Sie nach dem Erstellen eines Schnittstellen-VPC-Endpunkts [private DNS](#)-Hostnamen für den Endpunkt aktivieren, wird der ElastiCache-Standardendpunkt (<https://elasticache.Region.amazonaws.com>) in Ihren VPC-Endpunkt aufgelöst. Wenn Sie keine privaten DNS-Hostnamen aktiviert haben, stellt Amazon VPC einen DNS-Endpunktnamen bereit, den Sie im folgenden Format verwenden können:

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

Weitere Informationen finden Sie unter [Schnittstellen-VPC-Endpunkte \(AWS-PrivateLink\)](#) im Amazon-VPC-Benutzerhandbuch. ElastiCache unterstützt Aufrufe aller [API-Aktionen](#) innerhalb Ihrer 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 VPC-Endpunkten

Bevor Sie einen Schnittstellen-VPC-Endpunkt für Amazon-ElastiCache-API-Endpunkte einrichten, stellen Sie sicher, dass Sie die [Eigenschaften und Einschränkungen des Schnittstellenendpunkts](#) im Amazon VPC Benutzerhandbuch einsehen. Alle ElastiCache-API-Operationen, die für die Verwaltung von Amazon-ElastiCache-Ressourcen relevant sind, sind mit AWS PrivateLink über Ihre VPC verfügbar.

VPC-Endpunkttrichtlinien werden für ElastiCache-API-Endpunkte unterstützt. Standardmäßig ist der vollständige Zugriff auf ElastiCache-API-Operationen über den Endpunkt zulässig. Weitere Informationen finden Sie unter [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Erstellen eines Schnittstellen-VPC-Endpunkts für die ElastiCache-API

Sie können einen VPC-Endpunkt für die Amazon-ElastiCache-API mithilfe der Amazon-VPC-Konsole oder der AWS CLI erstellen. Weitere Informationen finden Sie unter [Erstellung eines Schnittstellenendpunkts](#) im Benutzerhandbuch für Amazon VPC.

Nachdem Sie einen Schnittstellen-VPC-Endpunkt erstellt haben, können Sie private DNS-Hostnamen für den Endpunkt aktivieren. Wenn Sie dies tun, wird der standardmäßige Amazon-ElastiCache-Endpunkt (<https://elasticache.Region.amazonaws.com>) auf Ihren VPC-Endpunkt aufgelöst. Für die AWS-Regionen China (Peking) und China (Ningxia) können Sie API-Anforderungen mit dem VPC-Endpunkt mittels `elasticache.cn-north-1.amazonaws.com.cn` für Peking bzw. `elasticache.cn-northwest-1.amazonaws.com.cn` für Ningxia senden. Weitere Informationen finden Sie unter [Zugriff auf einen Service über einen Schnittstellenendpunkt](#) im Benutzerhandbuch für Amazon VPC.

Erstellen einer VPC-Endpunkttrichtlinie für die Amazon-ElastiCache-API

Sie können eine Endpunkttrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff auf die ElastiCache-API steuert. 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 [Steuerung des Zugriffs auf Services mit VPC-Endpunkten](#) im Amazon-VPC-Benutzerhandbuch.

Example VPC-Endpunktrichtlinie für ElastiCache-API-Aktionen

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für die ElastiCache-API. Wenn diese Richtlinie an einen Endpunkt angefügt wird, gewährt sie Zugriff auf die aufgelisteten ElastiCache-API-Aktionen für alle Prinzipale auf allen Ressourcen.

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster",
      "elasticache:CreateSnapshot"
    ],
    "Resource": "*"
  }]
}
```

Example VPC-Endpunkt-Richtlinie, die den gesamten Zugriff über ein angegebenes AWS-Konto verweigert

Die folgende VPC-Endpunktrichtlinie verweigert dem AWS-Konto **123456789012** jeglichen Zugriff auf Ressourcen, die den Endpunkt verwenden. 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 (in der Regel private Subnetze), die Sie für Ihre in einer Amazon Virtual Private Cloud (VPC)-Umgebung ausgeführten selbst entworfenen Cluster festlegen können.

Wenn Sie einen selbst entworfenen Cluster in einer Amazon VPC erstellen, müssen Sie eine Subnetzgruppe verwenden. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen und diese mit Ihrem Knoten zu verknüpfen.

ElastiCache bietet eine standardmäßige IPv4-Subnetzgruppe oder Sie können eine neue erstellen. Für IPv6 müssen Sie eine Subnetzgruppe mit einem IPv6-CIDR-Block erstellen. Wenn Sie sich für dual stack (Dual-Stack) entscheiden, müssen Sie einen Discovery-IP-Typ auswählen, entweder IPv6 oder IPv4.

ElastiCache Serverless verwendet keine Subnetzgruppenressource, sondern nutzt 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 ElastiCache-Ressourcen zu verwalten.

Weitere Informationen zur Verwendung von Subnetzgruppen in einer Amazon-VPC-Umgebung finden Sie unter [Zugriff auf Ihren Cluster oder die 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 Zusammenstellung von Subnetzen, die Sie für Ihre Cache-Cluster in einer VPC festlegen können. Wenn Sie einen Cache in einer VPC starten, müssen Sie eine Cache-Subnetzgruppe auswählen. ElastiCache verwendet dann diese Cache-Subnetzgruppe, um den einzelnen Cache-Knoten im Cache IP-Adressen innerhalb des Subnetzes zuzuordnen.

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 auswählen, ist eine standardmäßige Subnetzgruppe verfügbar oder Sie können auswählen, eine neue zu erstellen. ElastiCache verwendet diese Subnetzgruppe, um ein Subnetz und IP-Adressen innerhalb dieses Subnetzes auszuwählen und diese mit Ihrem Knoten zu verknüpfen. Wenn Sie Dual-Stack oder IPv6 auswählen, werden Sie angewiesen, Dual-Stack- oder IPv6-Subnetze zu erstellen. Weitere Informationen zu Netzwerktypen finden Sie unter [Netzwerktyp](#). Weitere Informationen finden Sie unter [Erstellen eines Subnetzes in Ihrer VPC](#).

Die folgenden Verfahren zeigen Ihnen, wie Sie eine Subnetzgruppe mit dem Namen mysubnetgroup (Konsole), die AWS CLI und die ElastiCache-API 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-Managementkonsole 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 Feld VPC-ID Ihre Amazon VPC aus.
 - d. Standardmäßig werden alle Subnetze ausgewählt. Klicken Sie im Bereich Ausgewählte Subnetze auf Verwalten und wählen Sie die Availability Zones oder [Lokale Zonen](#) und IDs Ihrer privaten Subnetze aus. Klicken Sie dann auf Auswählen.
5. Klicken Sie in der angezeigten Bestätigungsmeldung auf Close.

Ihre neue Subnetzgruppe wird in der Liste Subnetzgruppen der ElastiCache-Konsole angezeigt. 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 AWS CLI-Thema [create-cache-subnet-group](#).

Zuweisen einer Subnetzgruppe zu einem Cache

Nach dem Erstellen einer Subnetzgruppe können Sie einen Cache in einer Amazon-VPC starten. Weitere Informationen finden Sie unter den folgenden Topics.

- Redis-Standalone-Cluster – Informationen zum Starten eines Redis-Clusters mit einem Knoten finden Sie unter [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#). Wählen Sie in Schritt 7.a (Advanced Redis Settings) eine VPC-Subnetzgruppe aus.
- Replikationsgruppe Redis (Cluster-Modus deaktiviert) – Informationen zum Starten einer Redis-Replikationsgruppe (Cluster-Modus deaktiviert) in einer VPC finden Sie unter [Erstellen einer Redis-Replikationsgruppe \(Clustermodus deaktiviert\) von Grund auf](#). Wählen Sie in Schritt 7.b (Advanced Redis Settings) eine VPC-Subnetzgruppe aus.
- Redis (Cluster-Modus aktiviert)-Replikationsgruppe – [Erstellen eines Redis-Clusters \(Clustermodus aktiviert\) \(Konsole\)](#). Wählen Sie in Schritt 6.i (Advanced Redis Settings) eine VPC-Subnetzgruppe aus.

Ändern einer Subnetzgruppe

Sie können die Beschreibung einer Subnetzgruppe sowie die Liste der Subnetz-IDs, die der Subnetzgruppe zugeordnet sind, ändern. 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 AWS Management Console an 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 AWS Management Console an 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 unter Verwendung von 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 Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. ElastiCache IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen 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](#)
- [Übersicht über die Verwaltung von Zugriffsberechtigungen auf Ihre ElastiCache-Ressourcen](#)

Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie tätig sind. 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 anzufordern müssen. Wenn Sie in 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 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. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer

Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM nutzen 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 IAM verwenden können, 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 als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. 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) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS, 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 zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, um Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. 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 \(MFA\) in 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 unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

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 [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges](#)

[Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) 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 einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten 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 [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. 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 oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **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 unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, 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 andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.

- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal 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 kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch](#).
- **Serviceübergreifender Zugriff** — Einige verwenden Funktionen in anderen. AWS-Services AWS-Services Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-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. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen 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 für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstbezogene 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 Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Auf Amazon EC2 ausgeführte Anwendungen** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie

ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren 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 AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

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.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern,

welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

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 und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-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 von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

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 ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen 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 über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service Control Policies (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 Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer 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 [Funktionsweise von SCPs](#) 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 unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

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 IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

So ElastiCache arbeitet Amazon mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf verwenden, sollten Sie sich darüber informieren ElastiCache, mit welchen IAM-Funktionen Sie arbeiten können. ElastiCache

IAM-Funktionen, die Sie mit Amazon verwenden können ElastiCache

| IAM-Feature | ElastiCache Unterstützung |
|--|---------------------------|
| Identitätsbasierte Richtlinien | Ja |
| Ressourcenbasierte Richtlinien | Nein |
| Richtlinienaktionen | Ja |
| Richtlinienressourcen | Ja |
| Bedingungsschlüssel für die Richtlinie | Ja |
| ACLs | Ja |
| ABAC (Tags in Richtlinien) | Ja |
| Temporäre Anmeldeinformationen | Ja |
| Hauptberechtigungen | Ja |
| Servicerollen | Ja |
| Service-verknüpfte Rollen | Ja |

Einen allgemeinen Überblick darüber, wie ElastiCache und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Identitätsbasierte Richtlinien für ElastiCache

Unterstützt Richtlinien auf Identitätsbasis. Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

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 anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-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 kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie 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 Prinzipalentsität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. 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 unter [Kontenübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Politische Maßnahmen für ElastiCache

Unterstützt Richtlinienaktionen

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 `Action` 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, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. 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 angeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. 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 finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben 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 auf was Zugriff 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 die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

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

| | |
|------------------|----|
| Unterstützt ACLs | Ja |
|------------------|----|

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit ElastiCache

| | |
|--|----|
| Unterstützt ABAC (Tags in Richtlinien) | Ja |
|--|----|

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. 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, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam 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 ABAC finden Sie unter [Was ist ABAC?](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

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, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services](#) , [finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort 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 Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. 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 in IAM](#).

Serviceübergreifende Prinzipal-Berechtigungen für ElastiCache

Unterstützt Forward Access Sessions (FAS) Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-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. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-

Service zu stellen. FAS-Anfragen 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 für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für ElastiCache

| | |
|---------------------------|----|
| Unterstützt Servicerollen | Ja |
|---------------------------|----|

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Durch das Ändern der Berechtigungen für eine Servicerolle kann die ElastiCache Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, ElastiCache wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für ElastiCache

| | |
|--------------------------------------|----|
| Unterstützt serviceverknüpfte Rollen | Ja |
|--------------------------------------|----|

Eine dienstbezogene 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 Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). 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 verfügen Benutzer und Rollen nicht über die Berechtigung, ElastiCache-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben über die AWS Management Console, die AWS Command Line Interface (AWS CLI) oder die AWS-API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von ElastiCache definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon ElastiCache](#) in der Referenz zur Serviceautorisierung.

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:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie AWS-kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS-verwaltete Richtlinien](#) oder [AWS-verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung 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 von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der ElastiCache Konsole

Um auf die Konsole von Amazon ElastiCache zugreifen zu können, müssen Sie über eine Mindestanzahl von Berechtigungen verfügen. Diese Berechtigungen müssen Ihnen das Auflisten und Anzeigen von Details zu den ElastiCache-Ressourcen in Ihrem AWS-Konto gestatten. 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.

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen weiterhin die ElastiCache-Konsole verwenden können, fügen Sie den Entitäten auch die von `ConsoleAccess ElastiCache` oder `ReadOnly-AWS` verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die 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 IAM auftreten können.

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 auszuführen. ElastiCache 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. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, 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 dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs auf einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen](#).
- 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 über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie im IAM-Benutzerhandbuch unter [Kontenübergreifender Ressourcenzugriff in IAM](#).

Zugriffskontrolle

Sie können über gültige Anmeldeinformationen verfügen, um Ihre Anfragen zu authentifizieren, aber ohne die entsprechenden Berechtigungen können Sie keine Ressourcen erstellen oder darauf zugreifen ElastiCache . 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.

- [Übersicht über die Verwaltung von Zugriffsberechtigungen auf Ihre ElastiCache-Ressourcen](#)
- [Verwenden identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#)

Übersicht über die Verwaltung von Zugriffsberechtigungen auf Ihre ElastiCache-Ressourcen

Jede AWS-Ressource ist Eigentum eines AWS-Kontos und die Berechtigungen für die Erstellung einer Ressource oder den Zugriff darauf werden durch Berechtigungsrichtlinien geregelt. Ein Kontoadministrator kann Berechtigungsrichtlinien an IAM-Identitäten (Benutzer, Gruppen und Rollen) anfügen. Darüber hinaus unterstützt Amazon ElastiCache auch das Anfügen von Berechtigungsrichtlinien an Ressourcen.

Note

Ein Kontoadministrator (oder Administratorbenutzer) ist ein Benutzer mit Administratorrechten. Weitere Informationen finden Sie unter [Bewährte Methoden für IAM](#) 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, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter [Erstellen einer Rolle für einen externen Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch.

- IAM-Benutzer:

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

Themen

- [Amazon-ElastiCache-Ressourcen und -Operationen](#)
- [Grundlegendes zum Eigentum an Ressourcen](#)

- [Verwalten des Zugriffs auf Ressourcen](#)
- [AWS-verwaltete Richtlinien für Amazon ElastiCache](#)
- [Verwenden identitätsbasierter Richtlinien \(IAM-Richtlinien\) 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](#)

Amazon-ElastiCache-Ressourcen und -Operationen

Eine Liste von ElastiCache-Ressourcentypen und deren ARNs finden Sie unter [Von Amazon ElastiCache definierte Ressourcen](#) in der Referenz zur Serviceautorisierung. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von Amazon ElastiCache definierte Aktionen](#).

Grundlegendes zum Eigentum an Ressourcen

Ein Ressourcenbesitzer ist das AWS-Konto, das die Ressource erstellt hat. Das bedeutet, dass der Ressourcenbesitzer das AWS-Konto der Haupt-Entität ist, die die Anforderung authentifiziert, über die die Ressource erstellt wurde. Eine Haupt-Entität kann das Stammkonto, ein IAM-Benutzer oder eine IAM-Rolle sein. Die Funktionsweise wird anhand der folgenden Beispiele deutlich:

- Angenommen, Sie verwenden die Stammkonto-Anmeldeinformationen Ihres AWS-Kontos, um einen Cache-Cluster zu erstellen. In diesem Fall ist ihr AWS-Konto Besitzer der Ressource. In ElastiCache ist die Ressource der Cache-Cluster.
- Angenommen, Sie erstellen einen IAM-Benutzer in Ihrem AWS-Konto und erteilen ihm die Berechtigung, einen Cache-Cluster zu erstellen. In diesem Fall kann der Benutzer einen Cache-Cluster erstellen. Besitzer der Cache-Cluster-Ressource ist jedoch das AWS-Konto, zu dem der Benutzer gehört.
- Angenommen, Sie erstellen eine IAM-Rolle in Ihrem AWS-Konto mit den Berechtigungen, einen Cache-Cluster zu erstellen. In diesem Fall kann jeder, der die Rolle übernehmen kann, einen Cache-Cluster erstellen. Besitzer der Cache-Cluster-Ressource ist das AWS-Konto, zu dem die Rolle gehört.

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

Dieser Abschnitt behandelt die Verwendung von IAM im Zusammenhang mit Amazon ElastiCache. Er enthält keine detaillierten Informationen über den IAM-Service. Eine umfassende IAM-Dokumentation finden Sie unter [Was ist IAM?](#) im IAM-Benutzerhandbuch. Für Informationen über die Syntax und Beschreibungen von [AWS-IAM-Richtlinien](#) lesen Sie die IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

An eine IAM-Identität angefügte Richtlinien werden als identitätsbasierte Richtlinien (oder IAM-Richtlinien) bezeichnet. An Ressourcen angehängte Richtlinien werden als ressourcenbasierte Richtlinien bezeichnet.

Themen

- [Identitätsbasierte Richtlinien \(IAM-Richtlinien\)](#)
- [Angaben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale](#)
- [Angaben von Bedingungen in einer Richtlinie](#)

Identitätsbasierte Richtlinien (IAM-Richtlinien)

Richtlinien können IAM-Identitäten angefügt werden. 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 sind die Berechtigungen für diesen Benutzer zum Erstellen einer ElastiCache-Ressource bestimmt, z. B. eines Cache-Clusters, einer Parametergruppe oder einer Sicherheitsgruppe.
- Einer Rolle eine Berechtigungsrichtlinie zuweisen (kontoübergreifende Berechtigungen gewähren) – Sie können einer IAM-Rolle eine identitätsbasierte Berechtigungsrichtlinie zuweisen, um kontoübergreifende Berechtigungen zu erteilen. Beispielsweise kann der Administrator in Konto A eine Rolle erstellen, um einem anderen AWS-Konto (z. B. Konto B) oder einem AWS-Service kontoübergreifende Berechtigungen zu erteilen. Dazu geht er folgendermaßen vor:

1. Der Administrator von Konto A erstellt eine IAM-Rolle und fügt ihr eine Berechtigungsrichtlinie an, die Berechtigungen für Ressourcen in Konto A erteilt.
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 nun Berechtigungen zur Übernahme der Rolle an alle Benutzer in Konto B delegieren. Dadurch können die Benutzer in Konto B Ressourcen in Konto A erstellen oder auf diese zugreifen. In manchen Fällen möchten Sie möglicherweise einem AWS-Service die 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 zum Delegieren von Berechtigungen mithilfe von IAM finden Sie unter [Zugriffsverwaltung](#) im IAM-Benutzerhandbuch.

Im Folgenden finden Sie eine Beispielrichtlinie, die es einem Benutzer ermöglicht, die DescribeCacheClusters-Aktion für Ihr AWS-Konto auszuführen. ElastiCache unterstützt auch die Identifizierung bestimmter Ressourcen mithilfe der Ressourcen-ARNs für API-Aktionen. (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 ElastiCache finden Sie unter [Verwenden identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#). Weitere Informationen zu Benutzern, Gruppen, Rollen und Berechtigungen finden Sie unter [Identitäten \(Benutzer, Gruppen und Rollen\)](#) im IAM-Benutzerhandbuch.

Angeben der Richtlinienelemente: Aktionen, Effekte, Ressourcen und Prinzipale

Für jede Amazon-ElastiCache-Ressource (siehe [Amazon-ElastiCache-Ressourcen und -Operationen](#)) definiert der Service eine Reihe von API-Vorgängen (siehe [Aktionen](#)). Zur Erteilung von Berechtigungen für diese API-Operationen definiert ElastiCache eine Reihe von Aktionen, die Sie in einer Richtlinie angeben können. Für die ElastiCache-Cluster-Ressource sind beispielsweise die folgenden Aktionen definiert: `CreateCacheCluster`, `DeleteCacheCluster` und `DescribeCacheCluster`. Für das Durchführen einer API-Operation können Berechtigungen für mehrere Aktionen erforderlich sein.

Grundlegende Richtlinienelemente:

- **Ressource** – In einer Richtlinie wird der Amazon-Ressourcenname (ARN) zur Identifizierung der Ressource verwendet, für die die Richtlinie gilt. Weitere Informationen finden Sie unter [Amazon-ElastiCache-Ressourcen und -Operationen](#).
- **Aktion** – Mit Aktionsschlüsselwörtern geben Sie die Ressourcenoperationen an, die Sie zulassen oder verweigern möchten. Abhängig von der angegebenen `Effect` erlaubt oder verweigert die `elasticache:CreateCacheCluster`-Berechtigung beispielsweise die Benutzerberechtigungen zum Ausführen des Amazon-ElastiCache-Vorgangs `CreateCacheCluster`.
- **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.
- **Prinzipal** – In identitätsbasierten Richtlinien (IAM-Richtlinien) ist der Benutzer, dem die Richtlinie zugewiesen ist, automatisch der 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 Syntax und zu Beschreibungen von IAM-Richtlinien finden Sie in der [AWS-IAM-Richtlinienreferenz](#) im IAM-Benutzerhandbuch.

Eine Tabelle mit allen Amazon-ElastiCache-API-Aktionen finden Sie unter [ElastiCache API-Berechtigungen: Referenz zu Aktionen, Ressourcen und Bedingungen](#).

Angeben von Bedingungen in einer Richtlinie

Beim Erteilen von Berechtigungen können Sie mithilfe der IAM-Richtliniensyntax die Bedingungen angeben, unter denen die Richtlinie wirksam werden soll. Beispielsweise kann festgelegt werden, dass eine Richtlinie erst ab einem bestimmten Datum gilt. Weitere Informationen zum Angeben von Bedingungen in einer Richtliniensyntax finden Sie im Thema [Bedingung](#) im IAM Benutzerhandbuch.

Bedingungen werden mithilfe vordefinierter Bedingungsschlüssel formuliert. Informationen zur Verwendung von ElastiCache-spezifischen Bedingungsschlüsseln finden Sie unter [Verwenden von Bedingungsschlüssel](#). Es gibt AWS-weite Bedingungsschlüssel, die Sie nach Bedarf verwenden können. Sie finden eine vollständige Liste der AWS-weiten Schlüssel unter [Verfügbare Schlüssel für Bedingungen](#) im IAM-Benutzerhandbuch.

AWS-verwaltete Richtlinien für Amazon ElastiCache

Eine von AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Von AWS verwaltete Richtlinien stellen Berechtigungen für viele häufige Anwendungsfälle bereit, damit Sie beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise nicht die geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [kundenverwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Die Berechtigungen, die in den von AWS verwalteten Richtlinien definiert sind, können nicht geändert werden. Wenn AWS Berechtigungen aktualisiert, die in einer von AWS verwalteten Richtlinie definiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am wahrscheinlichsten eine von AWS verwaltete Richtlinie, wenn ein neuer AWS-Service gestartet wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

AWS-verwaltete Richtlinie: ElastiCacheServiceRolePolicy

Sie können ElastiCacheServiceRolePolicy nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist an eine serviceverknüpfte Rolle angehängt, die ElastiCache die Durchführung von Aktionen in Ihrem Namen ermöglicht.

Diese Richtlinie ermöglicht es ElastiCache, AWS-Ressourcen in Ihrem Namen nach Bedarf im Rahmen des Managements Ihres Caches zu verwalten:

- `ec2` – verwaltet EC2-Netzwerkressourcen, die an Cache-Knoten angehängt werden sollen, darunter VPC-Endpunkte (für Serverless-Caches), Elastic-Netzwerk-Schnittstellen (ENIs) (für selbst entworfene Cluster) und Sicherheitsgruppen.
- `cloudwatch` – sendet Metrikdaten aus dem Service an CloudWatch aus.

- `outposts` – erlaubt die Erstellung von Cache-Knoten in AWS Outposts.

Sie finden die Richtlinie [ElastiCacheServiceRolePolicy](#) in der IAM-Konsole und [ElastiCacheServiceRolePolicy](#) im Referenzleitfaden zur AWS-verwalteten Richtlinie.

AWS-verwaltete Richtlinie: `AmazonElastiCacheFullAccess`

Sie können die `AmazonElastiCacheFullAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Prinzipalen über die AWS-Managementkonsole Vollzugriff auf ElastiCache:

- `elasticache` – Zugriff auf alle APIs.
- `iam` – Erstellen einer serviceverknüpften Rolle, die für den Servicebetrieb erforderlich ist.
- `ec2` – Beschreiben von abhängigen EC2-Ressourcen, die für die Cache-Erstellung erforderlich sind (VPC, Subnetz, Sicherheitsgruppe), und Ermöglichen der Erstellung von VPC-Endpunkten (für Serverless-Caches).
- `kms` – Erlauben der Verwendung von kundenverwalteten CMKs für Verschlüsselung im Ruhezustand.
- `cloudwatch` – Erlauben des Zugriffs 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 auszufüllen.
- `sns` – wird verwendet, um SNS-Themen für die Benachrichtigungsfunktion in der Konsole auszufüllen.

Sie finden die Richtlinie [AmazonElastiCacheFullAccess](#) in der IAM-Konsole und [AmazonElastiCacheFullAccess](#) im Referenzleitfaden zur AWS-verwalteten Richtlinie.

AWS-verwaltete Richtlinie: AmazonElastiCacheReadOnlyAccess

Sie können die AmazonElastiCacheReadOnlyAccess-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Prinzipalen über die AWS-Managementkonsole schreibgeschützten Zugriff auf ElastiCache:

- `elasticache` – Zugriff auf schreibgeschützte `Describe`-APIs.

Sie finden die Richtlinie [AmazonElastiCacheReadOnlyAccess](#) in der IAM-Konsole und [AmazonElastiCacheReadOnlyAccess](#) im Referenzleitfaden zur AWS-verwalteten Richtlinie.

ElastiCache-Updates für AWS-verwaltete Richtlinien

Zeigen Sie Details zu Aktualisierungen für AWS-verwaltete Richtlinien für ElastiCache seit Beginn der Änderungsverfolgung durch diesen Service an. Abonnieren Sie den RSS-Feed auf der Seite für den ElastiCache-Dokumentverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

| Änderung | Beschreibung | Datum |
|--|---|-------------------|
| AmazonElastiCacheFullAccess – Aktualisierung einer bestehenden Richtlinie | ElastiCache hat neue Berechtigungen hinzugefügt, um die Verwaltung von Serverless-Caches und die Nutzung aller Servicefunktionen über die Konsole zu ermöglichen. | 27. November 2023 |
| ElastiCacheServiceRolePolicy – Aktualisierung einer bestehenden Richtlinie | ElastiCache hat neue Berechtigungen hinzugefügt, um die Verwaltung von VPC-Endpunkten für Serverless-Cache-Ressourcen zu ermöglichen. | 27. November 2023 |
| ElastiCache hat mit der Verfolgung von Änderungen begonnen | ElastiCache hat mit der Verfolgung von Änderungen | 7. Februar 2020 |

| Änderung | Beschreibung | Datum |
|----------|---|-------|
| | n für seine AWS-verwalteten Richtlinien begonnen. | |

Verwenden identitätsbasierter Richtlinien (IAM-Richtlinien) für Amazon ElastiCache

In diesem Thema finden Sie Beispiele für identitätsbasierte Richtlinien, in denen ein Kontoadministrator den IAM-Identitäten (Benutzer, Gruppen und Rollen) Berechtigungsrichtlinien anfügen kann.

Important

Wir empfehlen Ihnen, zuerst die Themen zu lesen, in denen die grundlegenden Konzepte und Optionen zum Verwalten des Zugriffs auf Amazon ElastiCache-Ressourcen erläutert werden. Weitere Informationen finden Sie unter [Übersicht über die Verwaltung von Zugriffsberechtigungen auf Ihre ElastiCache-Ressourcen](#).

Dieses Thema besteht aus folgenden Abschnitten:

- [AWS-verwaltete Richtlinien für Amazon ElastiCache](#)
- [Beispiele für vom Kunden verwaltete Richtlinien](#)

Dies ist ein Beispiel für eine Berechtigungsrichtlinie.

```
{
  "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"
  }
]
```

Die Richtlinie enthält zwei Anweisungen:

- Die erste Anweisung gewährt Berechtigungen für die Aktionen von Amazon ElastiCache (`elasticache:Create*`, `elasticache:Describe*`, `elasticache:Modify*`)
- Die zweite Anweisung erteilt Berechtigungen für die IAM-Aktion (`iam:PassRole`) für den IAM-Rollennamen am Ende des Resource-Wertes.

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. Wird die Berechtigungsrichtlinie einer IAM-Rolle angefügt, erhält der in der Vertrauensrichtlinie der Rolle angegebene Prinzipal die Berechtigungen.

Eine Tabelle mit allen Amazon ElastiCache-API-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: einem Benutzer erlauben, die IAM-CreateServiceLinkedRole-API aufzurufen](#)). Oder Sie sollten eine serviceverknüpfte ElastiCache-Rolle erstellt haben.

In Kombination mit den Mindestberechtigungen, die für die Verwendung der Amazon ElastiCache-Konsole erforderlich sind, gewähren die Beispiel-Richtlinien in diesem Abschnitt zusätzliche Berechtigungen. Die Beispiele sind auch für die AWS-SDKs und AWS CLI relevant.

Anweisungen zum Einrichten von IAM-Benutzern und -Gruppen finden Sie unter [Erstellen Ihrer ersten IAM-Benutzer- und -Administratorengruppe](#) im IAM-Benutzerhandbuch.

Important

Testen Sie Ihre IAM-Richtlinien immer gründlich, bevor Sie sie in der Produktion verwenden. Einige ElastiCache-Aktionen, die einfach erscheinen, benötigen möglicherweise andere unterstützende Aktionen, 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`- und `List`-Aktionen zum Auffüllen von Konsolenlisten.

Beispiele

- [Beispiel 1: einem Benutzer schreibgeschützten Zugriff auf ElastiCache-Ressourcen erlauben](#)
- [Beispiel 2: einem Benutzer erlauben, allgemeine Aufgaben des ElastiCache-Systemadministrators auszuführen](#)
- [Beispiel 3: einem Benutzer erlauben, auf alle ElastiCache-API-Aktionen zuzugreifen](#)
- [Beispiel 4: einem Benutzer erlauben, die IAM-CreateServiceLinkedRole-API aufzurufen](#)
- [Beispiel 5: einem Benutzer erlauben, mithilfe der IAM-Authentifizierung eine Verbindung mit dem Serverless-Cache herzustellen](#)

Beispiel 1: einem Benutzer schreibgeschützten Zugriff auf ElastiCache-Ressourcen erlauben

Die folgende Richtlinie erteilt Berechtigungen für ElastiCache-Aktionen, die es einem Benutzer erlauben, 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: einem Benutzer erlauben, allgemeine Aufgaben des ElastiCache-Systemadministrators auszuführen

Zu den allgemeinen Aufgaben des Systemadministrators gehört das Ändern von Ressourcen. Ein Systemadministrator möchte möglicherweise auch Informationen zu den ElastiCache-Ereignissen erhalten. Die folgende Richtlinie gewährt einem Benutzer Berechtigungen zum Ausführen von ElastiCache-Aktionen für diese allgemeinen Systemadministratortasken. 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: einem Benutzer erlauben, auf alle ElastiCache-API-Aktionen zuzugreifen

Die folgende Richtlinie erlaubt 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: einem Benutzer erlauben, die IAM-CreateServiceLinkedRole-API aufzurufen

Die folgende Richtlinie ermöglicht dem Benutzer den Aufruf der IAM-CreateServiceLinkedRole-API. Es wird empfohlen, diese Art von Berechtigungsrichtlinie dem Benutzer zu erteilen, der veränderliche ElastiCache-Vorgänge aufruft.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateSLRAllows",
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "elasticache.amazonaws.com"
        }
      }
    }
  ]
}
```

Beispiel 5: einem Benutzer erlauben, mithilfe der IAM-Authentifizierung eine Verbindung mit dem Serverless-Cache herzustellen

Die folgende Richtlinie ermöglicht es jedem Benutzer, mithilfe der IAM-Authentifizierung zwischen 01-04-2023 und 30-06-2023 eine Verbindung mit jedem beliebigen Serverless-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 festlegen. Viele ElastiCache-API-Aktionen unterstützen einen Ressourcentyp, der je nach Verhalten der Aktion variiert. Jede IAM-Richtlinienanweisung erteilt die Berechtigung für eine Aktion, die auf eine 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 (*). Für viele API-Aktionen können Sie die Ressourcen einschränken, die ein Benutzer ändern kann. Hierzu geben Sie den Amazon-Ressourcennamen (ARN) einer Ressource oder ein ARN-Muster an, das mehreren Ressourcen entspricht. Zum Einschränken von Berechtigungen nach Ressource bestimmen Sie die Ressource unter Angabe des ARN.

Eine Liste von ElastiCache-Ressourcentypen und deren ARNs finden Sie unter [Von Amazon ElastiCache definierte Ressourcen](#) in der Referenz zur Serviceautorisierung. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von Amazon ElastiCache definierte Aktionen](#).

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 ElastiCache können Sie das `Condition`-Element einer JSON-Richtlinie verwenden, um Schlüssel im Anforderungskontext mit Schlüssel-Werten zu vergleichen, die Sie in Ihrer Richtlinie angeben. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#).

Eine Liste der ElastiCache-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon ElastiCache](#) in der Referenz zur Serviceautorisierung.

Eine Liste der globalen Bedingungsschlüssel finden Sie unter [AWSGlobale Bedingungskontextschlüssel](#).

Festlegung von Bedingungen: Verwenden von Bedingungsschlüsseln

Um eine differenzierte Steuerung zu implementieren, schreiben Sie eine IAM-Berechtigungs-Richtlinie, die Bedingungen zum Steuern eines Satzes einzelner Parameter für bestimmte Anforderungen festlegt. Anschließend wenden Sie die Richtlinie für die IAM-Benutzer, -Gruppen oder -Rollen an, die Sie mithilfe der IAM-Konsole erstellen.

Um eine Bedingung anzuwenden, fügen Sie die Bedingungsinformationen der IAM-Richtlinienanweisung hinzu. Im folgenden Beispiel geben Sie die Bedingung an, dass jeder erstellte selbst entworfene Cache-Cluster vom Knotentyp `cache.r5.large` ist.

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

```

    }
  }
]
}

```

Weitere Informationen finden Sie unter [Beispiele für Tag-basierte Zugriffssteuerungsrichtlinien](#).

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 zum Implementieren einer differenzierten Zugriffskontrolle für die zuvor aufgeführten ElastiCache-Parameter.

1. `elasticache:MaximumDataStorage`: Geben Sie den maximalen Datenspeicher eines Serverless-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:MaximumECPUPerSecond`: Geben Sie den maximalen Wert für ECPU pro Sekunde für einen Serverless-Cache an. Unter den angegebenen Bedingungen kann der Kunde keine Caches erstellen, die mehr als eine bestimmte Anzahl ECPU pro Sekunde ausführen 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:MaximumECPUPerSecond": "100000"
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

3. `elasticache:CacheNodeType`: Geben Sie an, welche Knotentypen 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:NumNodeGroups`: Erstellen Sie 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"
        }
      }
    }
  ]
}
```

5. `elasticache:ReplicasPerNodeGroup`: Geben Sie die Replikat 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"
      }
    }
  }
]
}

```

6. `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"
      }
    }
  }
]
}

```

7. `elasticache:EngineType`: Geben Sie nur die Verwendung der Redis-Engine 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:EngineType": "redis"
        }
    }
}
]
}

```

8. `elasticache:AtRestEncryptionEnabled`: Geben Sie an, dass Replikationsgruppen nur mit 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"
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

9. elasticache:TransitEncryptionEnabled

- a. Legen Sie den `elasticache:TransitEncryptionEnabled`-Bedingungsschlüssel für die [CreateReplicationGroup](#)-Aktion auf `false` fest, um anzugeben, dass Replikationsgruppen nur erstellt werden können, wenn TLS nicht verwendet wird:

```

{
  "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 in einer Richtlinie für die Aktion [CreateReplicationGroup](#) auf `false` festgelegt ist, ist eine `CreateReplicationGroup`-Anforderung nur zulässig, wenn TLS nicht verwendet wird (das heißt, wenn die Anforderung keinen `TransitEncryptionEnabled`-Parameter enthält, der auf `true` festgelegt ist und keinen `TransitEncryptionMode`-Parameter, der auf `required` festgelegt ist).

- b. Legen Sie den `elasticache:TransitEncryptionEnabled`-Bedingungsschlüssel für die [CreateReplicationGroup](#)-Aktion auf `true` fest, um anzugeben, dass Replikationsgruppen nur erstellt werden können, wenn TLS verwendet wird:

```
{
  "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 in einer Richtlinie für die Aktion [CreateReplicationGroup](#) auf `true` festgelegt ist, ist eine `CreateReplicationGroup`-Anforderung nur zulässig, wenn die Anforderung einen `TransitEncryptionEnabled`-Parameter enthält, der auf `true` festgelegt ist und einen `TransitEncryptionMode`-Parameter, der auf `required` festgelegt ist.

- c. Legen Sie `elasticache:TransitEncryptionEnabled` für die `ModifyReplicationGroup`-Aktion auf `true` fest, um anzugeben, dass Replikationsgruppen nur geändert werden können, wenn TLS verwendet wird:

```
{
  "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 in einer Richtlinie für die Aktion [ModifyReplicationGroup](#) auf `true` festgelegt ist, ist eine `ModifyReplicationGroup`-Anforderung nur zulässig, wenn die Anforderung einen `TransitEncryptionMode`-Parameter enthält, der auf `required` festgelegt ist. Der `TransitEncryptionEnabled`-Parameter, der auf `true` festgelegt ist, kann optional ebenfalls enthalten sein, ist in diesem Fall jedoch nicht erforderlich, um TLS zu aktivieren.

- 10 `elasticache:AutomaticFailoverEnabled`: Geben Sie an, dass Replikationsgruppen nur mit 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"
        }
      }
    }
  ]
}

```

11 `elasticache:MultiAZEnabled`: Geben Sie an, dass Replikationsgruppen nicht mit deaktiviertem Multi-AZ 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": "Deny",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:CreateReplicationGroup"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
    "Condition": {
        "Bool": {
            "elasticache:MultiAZEnabled": "false"
        }
    }
}
]
}

```

elasticache:ClusterModeEnabled: Geben Sie an, dass Replikationsgruppen nur mit aktiviertem Cluster-Modus 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"
        }
    }
}
]
}

```

13. `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"
        }
    }
}
]
}

```

14 `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"
            }
        }
    ]
}

```

15 `elasticache:KmsKeyId`: Geben Sie die Verwendung der kundenverwalteten AWS-KMS-Schlüssel an.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:KmsKeyId": "my-key"
        }
      }
    }
  ]
}

```

16. `elasticache:CacheParameterGroupName`: Geben Sie eine nicht standardmäßige Parametergruppe mit bestimmten Parametern aus einer Organisation in Ihren Clustern an. Sie können auch ein Benennungsmuster für Ihre Parametergruppen angeben oder das Löschen eines bestimmten Parametergruppennamens blockieren. Es folgt ein Beispiel, das die Verwendung nur von "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"
        }
      }
    }
  ]
}
```

17 elasticache:CreateCacheCluster: 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"
    ]
  }
}
]
}

```

18elasticache:CacheNodeType: Zulassen von 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"
      }
    }
  ]
}

```

```
    }  
  }  
}  
]  
}
```

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 AWS Identity and Access Management (IAM) [serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist ein spezieller Typ der IAM-Rolle, die direkt mit einem AWS-Service wie z. B. Amazon ElastiCache verknüpft ist. Serviceverknüpfte Amazon-ElastiCache-Rollen sind von Amazon ElastiCache vordefiniert. Sie enthalten alle Berechtigungen, die der Dienst benötigt, um AWS-Dienste im Namen Ihrer Cluster aufzurufen.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon ElastiCache einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Die Rollen existieren bereits in Ihrem AWS-Konto, sind aber mit Amazon-ElastiCache-Anwendungsfällen verknüpft und haben vordefinierte Berechtigungen. Nur Amazon ElastiCache kann diese Rollen übernehmen und nur diese Rollen können die vordefinierte Berechtigungsrichtlinie verwenden. Sie können die Rollen nur nach dem Löschen der zugehörigen Ressourcen löschen. Dies schützt Ihre Amazon ElastiCache-Ressourcen, da Sie nicht versehentlich die erforderlichen Berechtigungen für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Inhalt

- [Serviceverknüpfte Rollenberechtigungen für Amazon ElastiCache](#)
 - [Berechtigungen zum Erstellen einer serviceverknüpften Rolle](#)
- [Erstellen einer serviceverknüpften Rolle \(IAM\)](#)
 - [Erstellen einer serviceverknüpften Rolle \(IAM-Konsole\)](#)
 - [Erstellen einer serviceverknüpften Rolle \(IAM-CLI\)](#)
 - [Erstellen einer serviceverknüpften Rolle \(IAM-API\)](#)
- [Bearbeiten der Beschreibung einer serviceverknüpften Rolle für Amazon ElastiCache](#)
 - [Bearbeiten der Beschreibung einer serviceverknüpften Rolle \(IAM-Konsole\)](#)
 - [Bearbeiten der Beschreibung einer serviceverknüpften Rolle \(IAM-CLI\)](#)
 - [Bearbeiten der Beschreibung einer serviceverknüpften Rolle \(IAM-API\)](#)
- [Löschen einer serviceverknüpften Rolle für Amazon ElastiCache](#)
 - [Bereinigen einer serviceverknüpften Rolle](#)
 - [Löschen einer serviceverknüpften Rolle \(IAM-Konsole\)](#)
 - [Löschen einer serviceverknüpften Rolle \(IAM-CLI\)](#)
 - [Löschen einer serviceverknüpften Rolle \(IAM-API\)](#)

Serviceverknüpfte Rollenberechtigungen für Amazon ElastiCache

Berechtigungen zum Erstellen einer serviceverknüpften Rolle

So erlauben Sie einer IAM-Entität, eine serviceverknüpfte Rolle mit `AWS-ServiceRoleForElastiCache` zu erstellen

Die folgende Berechtigungsanweisung zu den Berechtigungen für diese IAM-Entität hinzufügen:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole",
    "iam:PutRolePolicy"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWSServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

So erlauben Sie einer IAM-Entität, eine serviceverknüpfte Rolle mit `AWS-ServiceRoleForElastiCache` zu löschen

Die folgende Berechtigungsanweisung zu den Berechtigungen für diese IAM-Entität hinzufügen:

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/elasticache.amazonaws.com/AWSServiceRoleForElastiCache*",
  "Condition": {"StringLike": {"iam:AWSServiceName": "elasticache.amazonaws.com"}}
}
```

Alternativ können Sie eine AWS-verwaltete Richtlinie verwenden, um vollen Zugriff auf Amazon ElastiCache bereitzustellen.

Erstellen einer serviceverknüpften Rolle (IAM)

Sie können eine serviceverknüpfte Rolle mithilfe der IAM-Konsole, der CLI oder API erstellen.

Erstellen einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Erstellen einer serviceverknüpften Rolle verwenden.

So erstellen Sie eine serviceverknüpfte Rolle (Konsole)

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich der IAM Console Roles. 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 Or select a service to view its use cases (Oder Service auswählen, um Anwendungsfälle anzuzeigen) ElastiCache aus.
5. Wählen Sie Weiter: Berechtigungen aus.
6. Beachten Sie unter Richtlinienname, 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).

Erstellen einer serviceverknüpften Rolle (IAM-CLI)

Sie können die IAM-Operationen der AWS Command Line Interface für das Erstellen einer serviceverknüpften Rolle verwenden. Diese Rolle kann die Vertrauensrichtlinie, sowie die enthaltenen Richtlinien enthalten, die der Service für die Zuweisung der Rolle benötigt.

So erstellen Sie eine serviceverknüpfte Rolle (CLI)

Führen Sie die folgenden Operationen aus:

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

Erstellen einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Erstellen einer serviceverknüpften Rolle verwenden. Diese Rolle kann die Vertrauensrichtlinie, sowie die enthaltenen Richtlinien enthalten, die der Service für die Zuweisung der Rolle benötigt.

So erstellen Sie eine serviceverknüpfte Rolle (API)

Verwenden Sie den [CreateServiceLinkedRole](#)-API-Aufruf. Geben Sie in der Anforderung einen Servicenamen im Format `elasticache.amazonaws.com` an.

Bearbeiten der Beschreibung einer serviceverknüpften Rolle für Amazon ElastiCache

Amazon ElastiCache erlaubt Ihnen nicht, die serviceverknüpfte Rolle `AWS-ServiceRoleForElastiCache` 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 jedoch die Beschreibung der Rolle mit IAM bearbeiten.

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-Konsole)

Sie können die IAM-Konsole für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So bearbeiten Sie die Beschreibung einer serviceverknüpften Rolle (Konsole)

1. Wählen Sie im Navigationsbereich der IAM Console Roles (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).

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-CLI)

Sie können die IAM-Operationen der AWS Command Line Interface für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (CLI)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie die AWS CLI für die IAM-Operation [get-role](#).

Example

```
$ aws iam get-role --role-name AWSServiceRoleForElastiCache
```

Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den CLI-Operationen zu beziehen. Wenn eine Rolle zum Beispiel folgenden ARN hat: `arn:aws:iam::123456789012:role/myrole`, verweisen Sie auf die Rolle als **myrole**.

2. Um die Beschreibung einer serviceverknüpften Rolle zu aktualisieren, verwenden Sie die AWS CLI für die IAM-Operation [update-role-description](#).

Für Linux, macOS oder Unix:

```
$ aws iam update-role-description \  
  --role-name AWSServiceRoleForElastiCache \  
  --description "new description"
```

Für Windows:

```
$ aws iam update-role-description ^  
  --role-name AWSServiceRoleForElastiCache ^  
  --description "new description"
```

Bearbeiten der Beschreibung einer serviceverknüpften Rolle (IAM-API)

Sie können die IAM-API für das Bearbeiten der Beschreibung einer serviceverknüpften Rolle verwenden.

So ändern Sie die Beschreibung einer serviceverknüpften Rolle (API)

1. (Optional) Um die aktuelle Beschreibung einer Rolle anzuzeigen, verwenden Sie die IAM-API-Operation [GetRole](#).

Example

```
https://iam.amazonaws.com/  
?Action=GetRole  
&RoleName=AWSServiceRoleForElastiCache  
&Version=2010-05-08  
&AUTHPARAMS
```

2. Um die Beschreibung einer Rolle zu aktualisieren, verwenden Sie die IAM-API-Operation [UpdateRoleDescription](#).

Example

```
https://iam.amazonaws.com/  
?Action=UpdateRoleDescription  
&RoleName=AWSServiceRoleForElastiCache  
&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 mit IAM löschen können, bestätigen Sie zunächst, dass der Rolle keine Ressourcen (Cluster oder Replikationsgruppen) zugeordnet sind.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich der IAM Console Roles. Wählen Sie dann den Namen (nicht das Kontrollkästchen) der AWS-Rolle ServiceRoleForElastiCache 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.

So löschen Sie Amazon-ElastiCache-Ressourcen, die AWS-ServiceRoleForElastiCache erfordern

- Informationen zum Löschen eines Clusters finden Sie unter:
 - [Verwendung von AWS Management Console](#)
 - [Verwendung der AWS CLI](#)
 - [Verwenden der 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 (IAM-Konsole)

Sie können die IAM-Konsole für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (Konsole)

1. Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.

2. Wählen Sie im Navigationsbereich der IAM Console Roles 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 Bestätigungsdiaologfeld die letzten Service-Zugriffsdaten, die zeigen, wann jede der ausgewählten Rollen zuletzt auf den AWS-Service 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 Benachrichtigungen in der IAM-Konsole an, um den Fortschritt der Löschung der serviceverknüpften Rolle zu überwachen. Da die Löschung der serviceverknüpften IAM-Rolle asynchron erfolgt, kann die Löschung nach dem Übermitteln der Rolle für die Löschung erfolgreich sein oder fehlschlagen. 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 serviceverknüpften Rolle (IAM-CLI)

Sie können die IAM-Operationen der AWS Command Line Interface für das Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (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 (ARN) in Ihrem Konto auf.

```
$ aws iam get-role --role-name role-name
```

Verwenden Sie den Rollennamen, nicht den ARN, um sich auf Rollen mit den CLI-Operationen zu beziehen. Wenn eine Rolle zum Beispiel den ARN `arn:aws:iam::123456789012:role/myrole` hat, verweisen Sie auf 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 Löschtaufgabe 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 IAM-API zum Löschen einer serviceverknüpften Rolle verwenden.

So löschen Sie eine serviceverknüpfte Rolle (API)

1. Um eine Löschanfrage für eine serviceverknüpfte Rolle zu übermitteln, rufen Sie [DeleteServiceLinkedRole](#) auf. Geben Sie in der Anforderung 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. Um den Status der Löschung zu überprüfen, rufen Sie [GetServiceLinkedRoleDeletionStatus](#) auf. Geben Sie in der Anforderung die DeletionTaskId an.

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 [Zugriffskontrolle](#) und 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 Amazon-API-Operationen und die entsprechenden Aktionen aufgeführt, für die Sie Berechtigungen zur Ausführung 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 kein Ressourcen-ARN vorhanden ist, ist die Ressource in der Richtlinie als Platzhalter (*) dargestellt.

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 Präfix `elasticache:` gefolgt vom Namen der API-Operation (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 Services im Rahmen mehrerer AWS Compliance-Programme wie SOC, PCI, FedRAMP und HIPAA.

Informationen darüber, ob ein in den Geltungsbereich bestimmter Compliance-Programme AWS-Service fällt, finden Sie unter [AWS-Services Umfang nach Compliance-Programm](#) unter [Umfang nach Compliance-Programm](#) AWS-Services . Wählen Sie aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter [herunterladen AWS Artifact](#). Weitere Informationen finden Sie unter [Berichte herunterladen 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 HIPAA-fähige Anwendungen erstellen AWS können.

 Note

AWS-Services Nicht alle sind HIPAA-fähig. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmapen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- [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 den 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)) zugeordnet.
- [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](#) — Auf diese AWS-Service Weise erhalten Sie 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

wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.

- [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:

- [FIPS-Endpunkte nach Service](#)
- [Service-Updates in ElastiCache](#)
- [AWS Cloud-Konformität](#)
- [Modell der übergreifenden Verantwortlichkeit](#)
- [AWS PCI DSS-Compliance-Programm](#)

Ausfallsicherheit in Amazon ElastiCache

Im Zentrum der globalen AWS Infrastruktur stehen die AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs). AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die mit Netzwerken mit geringer Latenz, hohem Durchsatz und hochredundanten Vernetzungen 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 über AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Neben der globalen AWS-Infrastruktur bietet Amazon ElastiCache verschiedene Funktionen zum Support Ihrer Datenausfallsicherheit und Backup-Anforderungen.

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 Redis](#)
- [Empfehlungen](#)

Minimieren von Ausfällen mit Redis

Wenn Sie die Redis-Engine verwenden, haben Sie folgende Möglichkeiten, um die Auswirkungen von Knoten- und Availability-Zone-Ausfällen möglichst gering zu halten.

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.

Sie haben folgende Möglichkeiten, um die Auswirkungen von Redis-Knotenausfällen auf selbst entworfenen Clustern möglichst gering zu halten:

Themen

- [Minimieren von Ausfällen: Redis-Replikationsgruppen](#)

Minimieren von Ausfällen: Redis-Replikationsgruppen

Eine Redis-Replikationsgruppe besteht aus einem einzelnen primären Knoten für Lese- und Schreibzugriffe sowie 1 bis 5 Replikationsknoten ausschließlich für Lesezugriffe. Wenn Daten in den Primärknoten geschrieben werden, werden diese asynchron auf die Lesereplikat-Knoten aktualisiert.

Wenn ein Lesereplikat ausfällt,

1. ElastiCache erkennt die fehlgeschlagene Read Replica.
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.

Redis Multi-AZ

Sie können Multi-AZ für Ihre Redis-Replikationsgruppen aktivieren. 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 der Ausfallzeiten ElastiCache bei Redis mit Multi-AZ](#).

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 sollten Sie die Knoten in Ihrer Replikationsgruppe in verschiedenen Availability Zones (AZs) 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 die Knoten jedoch auf mehrere AZs verteilen, führt ein AZ-Ausfall nur zum Verlust der Daten in den Knoten dieser AZ.

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.

Weitere Informationen zum Festlegen der Availability Zones für Redis-Knoten erhalten Sie unter [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#).

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Auswahl von Regionen und Availability Zones](#).

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

Sie sollten mehrere Knoten pro Shard implementieren und diese Knoten auf mehrere Availability Zones verteilen, um die Auswirkungen eines Knotenausfalls möglichst gering zu halten. Dies erfolgt automatisch für Serverless-Caches.

Für selbst entworfene Cluster empfehlen wir, Multi-AZ in Ihrer Replikationsgruppe zu aktivieren, sodass bei einem Ausfall des ElastiCache Primärknotens automatisch ein Failover auf ein Replikat durchgeführt wird.

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. Wenn Sie die Knoten gleichmäßig auf Ihre AZs verteilen, halten Sie die Auswirkungen im unwahrscheinlichen Fall eines AZ-Ausfalls möglichst gering. Dies erfolgt automatisch für Serverless-Caches.

Weitere Vorsichtsmaßnahmen

Wenn Sie Redis verwenden, sollten Sie darüber hinaus regelmäßige Sicherungen Ihres Clusters 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](#).

Sicherheit der Infrastruktur in AWS ElastiCache

Als verwalteter Service ist AWS ElastiCache durch die globalen AWS-Netzwerksicherheitsverfahren geschützt, die im Abschnitt „Sicherheit und Compliance“ im [AWS-Architekturzentrum](#) beschrieben sind.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf ElastiCache zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.2 oder höher unterstützen. Wir empfehlen TLS 1.3 oder höher. Clients müssen außerdem Verschlüsselungssammlungen mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. 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 entwickelte Cluster richten Sie ein vordefiniertes Wartungsfenster ein, damit diese Updates anwenden ElastiCache kann. 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 den ausgewählten ElastiCache Cluster auch in Echtzeit überwachen.

Verwaltung von Service-Updates

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.

Themen

- [Anwenden der Service-Updates](#)
- [Mithilfe der Konsole wird überprüft, ob Sie das neueste Service Update installiert haben AWS](#)
- [Stoppen der Service-Updates](#)

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 `available` oder `planned` 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, die API oder verwenden, 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
 - `Aktualisierungs-Typ`: 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
```

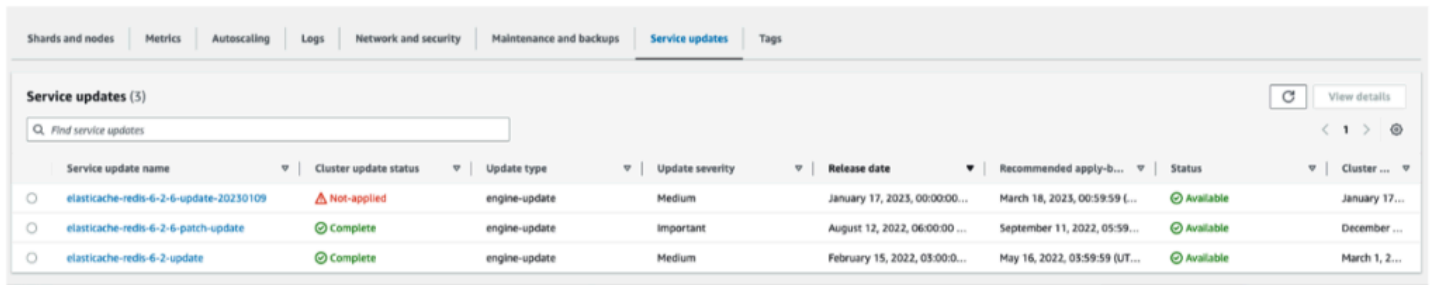
Weitere Informationen finden Sie unter [batch-apply-update-action](#).

Mithilfe der Konsole wird überprüft, ob Sie das neueste Service Update installiert haben AWS

Sie können überprüfen, ob auf Ihren ElastiCache Redis-Clustern das neueste Service-Update ausgeführt wird, indem Sie die folgenden Schritte ausführen:

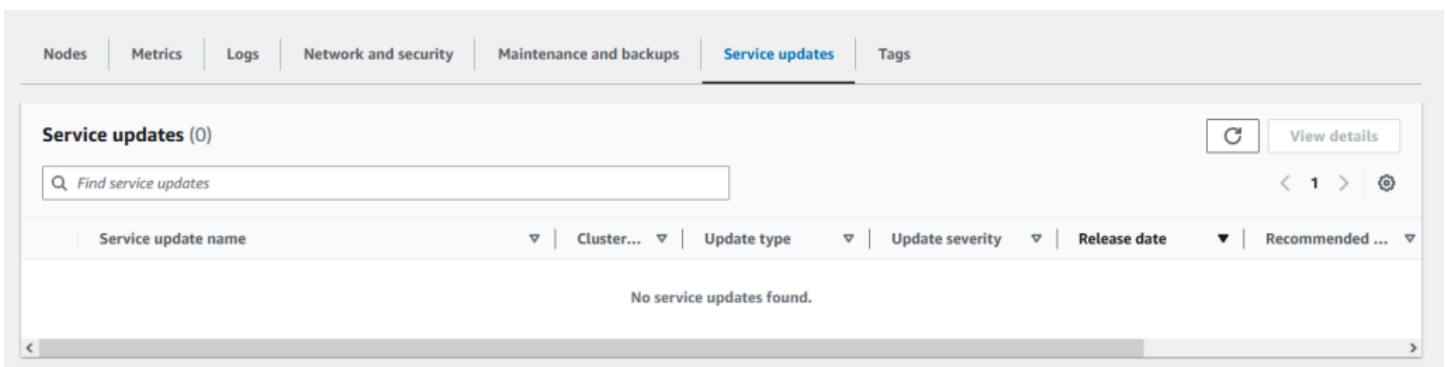
1. Wählen Sie auf der Seite Redis-Cluster einen geeigneten Cluster aus
2. Wählen Sie im Navigationsbereich Service Updates aus, um die entsprechenden Service-Updates für diesen Cluster, falls vorhanden, zu sehen.

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 ... |
|--|-----------------------|---------------|-----------------|-------------------------------|--------------------------------|-----------|---------------|
| elasticsearch-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... |
| elasticsearch-redis-6-2-6-patch-update | Complete | engine-update | Important | August 12, 2022, 06:00:00 ... | September 11, 2022, 05:59... | Available | December ... |
| elasticsearch-redis-6-2-update | Complete | engine-update | Medium | February 15, 2022, 03:00:0... | May 16, 2022, 05:59:59 (UT... | Available | March 1, 2... |

Wenn auf der Konsole „Keine Dienstupdates gefunden“ angezeigt wird, bedeutet dies, dass auf dem ElastiCache For-Redis-Cluster bereits das neueste Service-Update installiert wurde.



| Service update name | Cluster... | Update type | Update severity | Release date | Recommended ... |
|---------------------------|------------|-------------|-----------------|--------------|-----------------|
| No service updates found. | | | | | |

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](#).

Common Vulnerabilities and Exposures (CVE): Sicherheitslücken, die in ElastiCache für Redis behoben wurden

Common Vulnerabilities and Exposures (CVE) ist eine Liste von Einträgen für öffentlich bekannte Cybersicherheitsschwachstellen. 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 in ElastiCache Redis behoben wurden.

Wir empfehlen, immer auf die neueste Version von Redis zu aktualisieren, ElastiCache um vor bekannten Sicherheitslücken geschützt zu sein. Wenn Sie einen ElastiCache serverlosen Cache betreiben, werden CVE-Fixes automatisch auf Ihren Cache angewendet. Beim Betrieb von selbst entworfenen Clustern macht ElastiCache for Redis die PATCH-Komponente verfügbar. Wenn Sie beispielsweise Version 6.2.6 ElastiCache für Redis verwenden, ist die Hauptversion 6, die Nebenversion ist 2 und die Patch-Version ist 6. PATCH-Versionen sind für abwärtskompatible Bugfixes, Sicherheitskorrekturen und nicht funktionale Änderungen vorgesehen.

Sie können diese Seite verwenden, um zu überprüfen, ob eine bestimmte Version von ElastiCache for Redis eine Lösung für eine bestimmte Sicherheitslücke enthält. Wenn auf Ihrem ElastiCache for Redis-Cluster eine Version ohne den Sicherheitspatch ausgeführt wird, schlagen Sie in der folgenden Tabelle nach und ergreifen Sie Maßnahmen. Sie können entweder auf eine neuere Version ElastiCache für Redis aktualisieren, die den Fix enthält, oder wenn Sie eine Version ElastiCache für Redis verwenden, die den Fix enthält, stellen Sie anhand von sicher, dass Sie das neueste Service-Update installiert haben. [Verwaltung von Service-Updates](#) Weitere Informationen zu den unterstützten Versionen der Redis-Engine und ElastiCache zur Durchführung eines Upgrades finden Sie unter. [Engine-Versionen und Upgrades](#)

Note

- Wenn ein CVE in einer Version ElastiCache für Redis adressiert wird, bedeutet dies, dass es auch in den neueren Versionen adressiert wird. Wenn also beispielsweise eine Sicherheitslücke in ElastiCache der Redis-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 for Redis-Cluster installiert haben müssen, auf dem die angegebene ElastiCache for Redis-Version ausgeführt wird, ElastiCache um die Sicherheitslücke zu beheben.

Weitere Informationen darüber, wie Sie überprüfen können, ob Sie das neueste Service-Update für die Redis-Version installiert haben, auf der ElastiCache Ihr Cluster ausgeführt wird, finden Sie unter. [Verwaltung von Service-Updates](#)

| ElastiCache für die Redis-Version | Adressierte CVEs |
|-----------------------------------|---|
| Redis 6.0.5 | CVE-2022-24735 *, CVE-2022-24736 * |
| Redis 6.2.6 | CVE-2022-24834 *, CVE-2022-35977 *, CVE-2022-36021 *, CVE-2022-24735 , CVE-2022-24736 |
| Redis 7.0.7 | CVE-2023-41056 *, CVE-2022-24834 *, CVE-2022-35977 , CVE-2022-36021 , CVE-2022-24735 , CVE-2022-24736 |
| Redis 7.1.0 | CVE-2023-41056 , CVE-2022-24834 , CVE-2022-35977 , CVE-2022-36021 , CVE-2022-24735 , CVE-2022-24736 |

Protokollierung und Überwachung in Amazon ElastiCache

Damit Sie Ihren Cache verwalten können, müssen Sie wissen, wie Ihre Caches funktionieren. ElastiCache generiert Metriken, die zur Überwachung Ihrer Cache-Leistung in Amazon CloudWatch Logs veröffentlicht werden. Darüber hinaus generiert ElastiCache Ereignisse, wenn signifikante Änderungen an Ihren Cache-Ressourcen auftreten (z. B. wenn ein neuer Cache erstellt oder ein Cache gelöscht wird).

Themen

- [Serverless-Metriken und -Ereignisse](#)
- [Metriken und Ereignisse für selbst entworfene Cluster](#)
- [Protokollieren von Amazon-ElastiCache-API-Aufrufen mithilfe von AWS CloudTrail](#)

Serverless-Metriken und -Ereignisse

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

Der AWS/ElastiCache-Namespace umfasst die folgenden CloudWatch-Metriken für Ihre Serverless-Redis-Caches.

| Metrik | Beschreibung | Einheit |
|----------------------------|--|---------|
| BytesUsedForCache | Die Gesamtzahl der von den in Ihrem Cache gespeicherten Daten verwendeten Bytes. | Bytes |
| ElastiCacheProcessingUnits | Die Gesamtzahl der ElastiCacheProcessingUnits (ECPUs), | Anzahl |

| Metrik | Beschreibung | Einheit |
|-------------------------------|--|---------------|
| | die von den in Ihrem Cache ausgeführten Anfragen verbraucht werden | |
| 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 |
| ThrottledCmds | Die Anzahl der Anfragen, die von ElastiCache gedrosselt wurden, weil der Workload schneller skaliert wurde, als ElastiCache skalieren kann. | Anzahl |

| Metrik | Beschreibung | Einheit |
|------------------------------|--|---------|
| NewConnections | Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind. | Anzahl |
| Currltems | Anzahl der Elemente im Cache. | Anzahl |
| CurrVolatileItems | 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 Redis-Verbindungen. Weitere Informationen über Authentifizieren mit IAM finden Sie im Benutzerhandbuch. | Anzahl |
| IamAuthenticationThrottling | Die Gesamtzahl der gedrosselten IAM-authentifizierten Redis AUTH- oder HELLO-Anfragen. Weitere Informationen über Authentifizieren mit IAM finden Sie im Benutzerhandbuch. | Anzahl |

| Metrik | Beschreibung | Einheit |
|------------------------------|---|---------|
| 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 mit dem AUTH-Befehl bei Redis zu authentifizieren. Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen. | Anzahl |
| 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 außerdem die folgenden Metriken auf Befehlsebene aus. Für jeden Befehlstyp gibt ElastiCache die Gesamtzahl der Befehle und die Anzahl der von diesem Befehlstyp verbrauchten ECPUs aus.

| Metrik | Beschreibung | Einheit |
|--------------------------|---|---------|
| EvalBasedCmds | Die Anzahl der get-Befehle, die der Cache erhalten hat. | Anzahl |
| EvalBasedCmdsECPUs | ECPUs, die von eval-basierten Befehlen verbraucht werden. | Anzahl |
| GeoSpatialBasedCmds | Die Gesamtzahl der Befehle für raumbezogene Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet. Die Ableitung erfolgt durch Summieren aller Befehle des Geo-Typs: <code>geoadd</code> , <code>geodist</code> , <code>geohash</code> , <code>geopos</code> , <code>georadius</code> und <code>georadiusbymember</code> . | Anzahl |
| GeoSpatialBasedCmdsECPUs | ECPUs, die von geodatenbasierten Befehlen verbraucht werden. | Anzahl |
| GetTypeCmds | Die Gesamtzahl der schreibgeschützten Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle schreibgeschützten Befehle (<code>get</code> , <code>hget</code> , <code>scard</code> , <code>lrange</code> usw.) summiert werden | Anzahl |
| GetTypeCmdsECPUs | Von Lesebefehlen verbrauchte ECPUs. | Anzahl |
| HashBasedCmds | Gesamtanzahl der Hash-basierten Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, | Anzahl |

| Metrik | Beschreibung | Einheit |
|---------------------------|--|---------|
| | indem alle Befehle, die auf eine oder mehrere Hashes wirken, summiert werden (hget, hkeys, hvals, hdel usw.). | |
| HashBasedCmdsECPUs | ECPUs, die von Hash-basierten Befehlen verbraucht werden. | Anzahl |
| HyperLogLogBasedCmds | Die Gesamtanzahl der auf HyperLogLog basierenden Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle <code>pf</code> -Befehle (<code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> usw.) summiert werden. | Anzahl |
| HyperLogLogBasedCmdsECPUs | ECPUs, die von auf HyperLogLog basierenden Befehlen verbraucht werden. | Anzahl |
| JsonBasedCmds | Die Gesamtzahl der JSON-Befehle, einschließlich Lese- und Schreibbefehlen. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle JSON-Befehle summiert werden, die auf JSON-Schlüssel einwirken. | Anzahl |
| JsonBasedCmdsECPUs | Die von allen JSON-Befehlen, einschließlich Lese- und Schreibbefehlen, verbrauchten ECPUs. | Anzahl |

| Metrik | Beschreibung | Einheit |
|-----------------------|---|---------|
| JsonBasedGetCmds | Gesamtanzahl der JASON-Schreibschutzbefehle. Dieser Wert wird aus der Redis-Statistik commandstats abgeleitet, indem alle JSON-Lesebefehle summiert werden, die auf JSON-Schlüssel einwirken. | Anzahl |
| JsonBasedGetCmdsECPUs | ECPUs, die von schreibgeschützten JSON-Befehlen verbraucht werden. | Anzahl |
| JsonBasedSetCmds | Gesamtanzahl der JASON-Schreibbefehle. Dieser Wert wird aus der Redis-Statistik commandstats abgeleitet, indem alle JSON-Schreibbefehle summiert werden, die auf JSON-Schlüssel einwirken. | Anzahl |
| JsonBasedSetCmdsECPUs | ECPUs, die von JSON-Schreibbefehlen verbraucht werden. | Anzahl |
| KeyBasedCmds | Gesamtanzahl der schlüsselbasierten Befehle. Dieser Wert wird aus der Redis-Statistik commandstats abgeleitet, indem alle Befehle summiert werden, die über mehrere Datenstrukturen hinweg auf mindestens einen Schlüssel einwirken (del, expire, rename usw.). | Anzahl |

| Metrik | Beschreibung | Einheit |
|---------------------|---|---------|
| KeyBasedCmdsECPUs | ECPUs, die von schlüsselbasierten Befehlen verbraucht werden. | Anzahl |
| ListBasedCmds | Gesamtanzahl der listenbasierten Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle Befehle summiert werden, die sich auf eine oder mehrere Listen auswirken (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> usw.). | Anzahl |
| ListBasedCmdsECPUs | ECPUs, die von listenbasierten Befehlen verbraucht werden. | Anzahl |
| NonKeyTypeCmds | Gesamtanzahl der nicht schlüsselbasierten Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle Befehle summiert werden, die nicht auf einen Schlüssel einwirken, z. B. <code>acl</code> , <code>dbsize</code> oder <code>info</code> . | Anzahl |
| NonKeyTypeCmdsECPUs | ECPUs, die von nicht schlüsselbasierten Befehlen verbraucht werden. | Anzahl |

| Metrik | Beschreibung | Einheit |
|----------------------|---|---------|
| PubSubBasedCmds | Die Gesamtzahl der Befehle für die Pub/Sub-Funktionalität. Dieser Wert wird aus den Redis-Statistiken <code>commandstats</code> abgeleitet, indem alle für die Pub/Sub-Funktionalität verwendeten Befehle summiert werden: <code>psubscribe</code> , <code>publish</code> , <code>pubsub</code> , <code>punsubscribe</code> , <code>ssubscribe</code> , <code>sunsubscribe</code> , <code>spublish</code> , <code>subscribe</code> , and <code>unsubscribe</code> . | Anzahl |
| PubSubBasedCmdsECPUs | ECPUs, die von Pub/Sub-basierten Befehlen verbraucht werden. | Anzahl |
| SetBasedCmds | Gesamtanzahl der Set-basierten Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle Befehle, die sich auf ein oder mehrere Sets auswirken, summiert werden (<code>scard</code> , <code>sdiff</code> , <code>sadd</code> , <code>sunion</code> usw.). | Anzahl |
| SetBasedCmdsECPUs | ECPUs, die von setbasierten Befehlen verbraucht werden. | Anzahl |

| Metrik | Beschreibung | Einheit |
|-------------------------|--|---------|
| SetTypeCmds | Gesamtanzahl der Schreibbe fehle. Dieser Wert wird von der Redis-Statistik commandstats abgeleitet, indem alle Arten von veränderlichen Befehlen, die auf Daten wirken, summiert werden (set, hset, sadd, lpop usw.). | Anzahl |
| SetTypeCmdsECPUs | ECPUs, die von Schreibbe fehlen verbraucht werden. | Anzahl |
| SortedSetBasedCmds | Gesamtanzahl der Sorted Set-basierten Befehle. Dieser Wert wird von der Redis-Statistik commandstats abgeleitet, indem alle Befehle, die sich auf ein oder mehrere sortierte Sets auswirken, summiert werden (zcount, zrange, zrank, zadd usw.). | Anzahl |
| SortedSetBasedCmdsECPUs | ECPUs, die von sortierba sierten Befehlen verbraucht werden. | Anzahl |
| StringBasedCmds | Gesamtanzahl der Zeichenfo lge-basierten Befehle. Dieser Wert wird aus der Redis-Statistik commandstats abgeleitet, indem alle Befehle, die sich auf eine oder mehrere Zeichenfolgen auswirken , summiert werden (strlen, setex, setrange usw.). | Anzahl |

| Metrik | Beschreibung | Einheit |
|----------------------|--|---------|
| StringBasedCmdsECPUs | ECPUs, die von zeichenfo lgenbasierten Befehlen verbraucht werden. | Anzahl |
| StreamBasedCmds | Die Gesamtanzahl Stream-ba sierter Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle Befehle summiert werden, die sich auf einen oder mehrere Stream-Da tentypen auswirken (<code>xrange</code> , <code>xlen</code> , <code>xadd</code> , <code>xdel</code> usw.). | Anzahl |
| StreamBasedCmdsECPUs | ECPUs, die von Stream-ba sierten Befehlen verbraucht werden. | Anzahl |

Serverless-Cache-Ereignisse

ElastiCache protokolliert Ereignisse, die sich auf Ihren Serverless-Cache beziehen. Diese Informationen beinhalten Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mit der ElastiCache-Konsole, dem AWS CLI-Befehl `describe-events` oder der ElastiCache-API-Aktion `DescribeEvents` einfach aus dem Protokoll abrufen.

Sie können auswählen, ob Sie ElastiCache-Ereignisse mithilfe von Amazon EventBridge überwachen, aufnehmen, transformieren und darauf reagieren möchten. Weitere Informationen finden Sie unter Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/>.

Anzeigen von ElastiCache-Ereignissen (Konsole)

So zeigen Sie Ereignisse mithilfe der ElastiCache-Konsole an:

1. Melden Sie sich bei der AWS Management Console an 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 Uhrzeit (GMT) 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.

Anzeigen von ElastiCache-Ereignissen (AWS CLI)

Verwenden Sie zum Generieren einer Liste der ElastiCache-Ereignisse mithilfe der AWS CLI 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 VPC-Endpoint zu erstellen. |
| 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 Kontingentlimit 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 inkompatiblen 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> ist nicht vorhanden. |

| detail-type | Beschreibung | Einheit | Quelle | Fehlermeldung |
|---------------------------------|--------------|---------|------------------|---|
| Cache-Erstellung fehlgeschlagen | Cache arn | Ausfall | Serverless-Cache | <p>Der Cache <cache-name> konnte nicht erstellt werden. Die Datenwiederherstellung aus dem Snapshot ist fehlgeschlagen, weil <reason>.</p> <p>Gründe für das Fehlschlagen:</p> <ul style="list-style-type: none"> • Die Datei konnte nicht aus S3 abgerufen werden. • Das erwartete md5 stimmt nicht mit dem tatsächlichen md5 überein. • Die Version der bereitgestellten RDB-Datei wird nicht unterstützt. |

Ereignisse beim Aktualisieren eines Serverless-Caches

| detail-type | Liste der Ressourcen | Kategorie | Quelle | Fehlermeldung |
|-------------------------------------|----------------------|------------------------|------------------|---|
| Cache wurde aktualisiert | Cache arn | Konfigurationsänderung | Serverless-Cache | SecurityGroups wurden für den Cache <cache-name> 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 Redis-Version ist in der Benutzergruppe <user-group-name> vorhanden. |
| Cache-Aktualisierung fehlgeschlagen | Cache arn | Konfigurationsänderung | Serverless-Cache | Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups-Update 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. Die SecurityGroups-Aktualisierung ist aufgrund unzureichender Berechtigungen fehlgeschlagen. |
| Cache-Aktualisierung fehlgeschlagen | Cache arn | Konfigurationsänderung | Serverless-Cache | Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. Die SecurityGroups-Aktualisierung ist fehlgeschlagen, weil die SecurityGroups ungültig sind. |

Ereignisse beim Löschen von Serverless-Caches

| 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 Nutzung des Serverless-Caches

| 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. |

Snapshot-Ereignisse für den Serverless-Cache

| 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 <cache-name> konnte nicht erstellt werden. Die Erstellung des Snapshots <snapshot-name> ist mit dem kundenverwalteten Schlüssel <key-id> fehlgeschlagen <reason>.</p> <p>Meldungen zur Fehlerursache:</p> <ul style="list-style-type: none"> • Der kundenverwaltete Schlüssel wurde deaktiviert. • Der kundenverwaltete Schlüssel konnte nicht |

| detail-type | Liste der Ressourcen | Kategorie | Quelle | Fehlermeldung |
|---|-----------------------------------|----------------|----------------------------------|--|
| | | | | <p>gefunden werden.</p> <ul style="list-style-type: none"> • Das Zeitlimit für die Anfrage wurde überschritten. |
| <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 <cache-name> konnte nicht erstellt werden. Die Erstellung des Snapshots <snapshot-name> ist fehlgeschlagen, weil <reason>.</p> <p>Standardursache:</p> <ul style="list-style-type: none"> • Es liegt ein interner Fehler vor. |

| 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 keine Berechtigungen für den Bucket 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 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. |

Metriken und Ereignisse für selbst entworfene Cluster

In diesem Abschnitt werden die Metriken, Ereignisse und Protokolle beschrieben, die Sie bei der Arbeit mit selbst entworfenen Clustern erwarten können.

Themen

- [Metriken für selbst entworfene Cluster](#)
- [Ereignisse für selbst entworfene Cluster](#)
- [Protokollzustellung](#)
- [Überwachung der Nutzung mithilfe von CloudWatch-Metriken](#)
- [Amazon-SNS-Überwachung von ElastiCache-Ereignissen](#)

Metriken für selbst entworfene Cluster

Wenn Sie Cluster selbst entwerfen, gibt ElastiCache Metriken auf jeder Knotenebene aus, einschließlich Metriken auf Host-Ebene sowie 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 Redis](#).

Ereignisse für selbst entworfene Cluster

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 der AWS CLI oder mithilfe von Amazon Simple Notification Service (SNS) anzeigen. Ereignisse für selbst entworfene Cluster werden nicht in 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 mit der ElastiCache-Konsole, dem AWS CLI-Befehl `describe-events` oder der ElastiCache-API-Aktion `DescribeEvents` einfach aus dem Protokoll abrufen.

Anzeigen von ElastiCache-Ereignissen (Konsole)

Das folgende Verfahren zeigt Ereignisse mithilfe der ElastiCache-Konsole an.

So zeigen Sie Ereignisse mithilfe der ElastiCache-Konsole an

1. Melden Sie sich bei der AWS Management Console an 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 Uhrzeit (GMT) 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.

Anzeigen von ElastiCache-Ereignissen (AWS CLI)

Verwenden Sie zum Generieren einer Liste der ElastiCache-Ereignisse mithilfe der AWS CLI 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 Amazon-SNS-Benachrichtigungen aus. Weitere Informationen zu Ereignisdetails finden Sie unter [Anzeigen von ElastiCache-Ereignissen](#).

| 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 wegen nicht genügend 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 Netzwerkstatus | ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i> | Es wurde versucht, einen neuen Cache-Cluster in einer nicht vorhandenen Virtual Private 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">• Die Liste der Cache-Sicherheitsgruppen, die für den Cache-Cluster autorisiert sind, wurde geändert.• Mindestens eine neue EC2-Sicherheitsgruppe wurde für eine der Cache-Sicherheitsgruppen autorisiert, die dem Cache-Cluster zugeordnet sind.• Mindestens eine EC2-Sicherheitsgruppe wurde aus den Cache-Sicherheitsgruppen entfernt, die dem Cache-Cluster zugeordnet sind. |

| Ereignisname | Fehlermeldung | Beschreibung |
|-------------------------------------|---|---|
| Elasticache:CacheNoderePlaceStarted | ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i> | <p>ElastiCache hat erkannt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, schwach oder nicht erreichbar ist, und hat damit 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> Note</p><p>Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert.</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 nicht mehr, selbst nachdem ElastiCache den Cache-Knoten ersetzt hat. 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 erkannt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, schwach oder nicht erreichbar ist, und hat den Cache-Knoten ersetzt.</p> <div data-bbox="1068 541 1507 856"><p> Note</p><p>Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert.</p></div> <p>In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Clients-Bibliotheken verwenden den Cache-Knoten jedoch möglicherweise nicht mehr, selbst nachdem ElastiCache den Cache-Knoten ersetzt hat. 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 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 Redis) | 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> | Für einen zu ersetzenden Knoten in Ihrem Cluster wurde eine spätere Ersetzung im neuen Fenster geplant, das in der Benachrichtigung angegeben wurde. Weitere Informationen zu den möglichen Aktionen erhalten Sie unter Ersetzen von Knoten . |
| ElastiCache:NodeReplacementScheduled | ElastiCache:NodeReplacementScheduled : <i>cluster-name</i> | Ein Knoten in Ihrem Cluster soll während des in der Benachrichtigung beschriebenen Fensters ersetzt werden. Weitere Informationen zu den möglichen Aktionen erhalten Sie unter Ersetzen von Knoten . |
| 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 Redis) | ElastiCache:SnapshotComplete : <i>cluster-name</i> | Ein Cache-Snapshot wurde erfolgreich abgeschlossen. |
| ElastiCache:SnapshotFailed (nur Redis) | 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 DescribeSnapshots), ist dessen Status failed. |

Protokollzustellung

Note

Redis Slow Log wird für Redis-Cache-Cluster und -Replikationsgruppen mit Engine ab Version 6.0 unterstützt.

Redis Engine Log wird für Redis-Cache-Cluster und -Replikationsgruppen mit Engine ab Version 6.2 unterstützt.

Mit der Protokollzustellung können Sie Redis [SLOWLOG](#) oder Redis Engine Log an eines von zwei Zielen streamen:

- Amazon Data Firehose
- Amazon CloudWatch -Protokolle

Sie aktivieren und konfigurieren die Protokollzustellung, wenn Sie einen Cluster mit ElastiCache APIs erstellen oder ändern. Jeder Protokolleintrag wird in einem von zwei Formaten an das angegebene Ziel übermittelt: JSON oder TEXT.

Eine feste Anzahl von langsamen Protokolleinträgen wird regelmäßig von der Redis-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 Bereitstellungskonfigurationen jederzeit über die AWS Konsole oder eine der Änderungs-APIs, entweder `modify-cache-cluster` oder `modify-replication-group`, ändern [modify-cache-cluster](#) oder die Protokollzustellung deaktivieren [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 übermittelt werden. Weitere Informationen finden Sie im Abschnitt Vended Logs unter [Amazon CloudWatch -Preise](#).

Inhalt eines langsamen Protokolleintrags

ElastiCache für Redis 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` wobei `foo` der Schlüssel und der Wert `bar` ist. ElastiCache für Redis ersetzt den tatsächlichen Schlüsselnamen und -wert durch `(2 more arguments)`, um zu vermeiden, dass sensible Daten offengelegt werden.
- `ClientAddress` – Client-IP-Adresse und Port
- `ClientName` – Clientname, wenn über den `CLIENT SETNAME` Befehl festgelegt

Inhalte eines Engine-Protokolleintrags

Das ElastiCache for Redis Engine Log enthält die folgenden Informationen:

- `CacheClusterId` – Die ID des Cache-Clusters
- `CacheNodeId` – Die ID des Cache-Knotens
- `Protokollebene` – `LogLevel` kann eine der folgenden Optionen haben: `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. Dies kann einer der folgenden sein: „M“ für „Primär“, „S“ für „Replik“, „C“ für „An RDB/AOF arbeitender untergeordneter Schreibvorgang“ oder „X“ für „Sentinel“.
- `Nachricht` – Redis-Engine-Protokollnachricht.

Berechtigungen zum Konfigurieren der Protokollierung

Sie müssen die folgenden IAM-Berechtigungen in Ihre IAM-Benutzer-/Rollenrichtlinie einschließen:

- 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

Das langsame Protokoll unterstützt sowohl JSON als auch TEXT

Im Folgenden sehen Sie ein Beispiel für das JSON-Format:

```
{
  "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 sehen Sie ein Beispiel für das TEXT-Format:

```
logslowxxxxmsxj,0001,1605631822,30,GET ... (1 more
arguments),192.168.12.104:55452,logslowxxxxmsxj##
```

Engine-Protokoll

Das Engine-Protokoll unterstützt sowohl JSON als auch TEXT

Im Folgenden sehen Sie ein Beispiel für das JSON-Format:

```
{
  "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 sehen Sie ein Beispiel für das TEXT-Format:

```
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 Protokollierungsziele

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

- [Amazon CloudWatch -Protokolle](#)
- [Amazon Data Firehose](#)

Amazon CloudWatch -Protokolle

- Sie geben eine CloudWatch Protokollgruppe an, an die die Protokolle übermittelt werden.
- Protokolle aus mehreren Redis-Clustern und -Replikationsgruppen können an dieselbe Protokollgruppe übertragen 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 - CloudWatch Protokollen

Sie müssen über die folgenden Berechtigungseinstellungen verfügen, damit ElastiCache Redis Protokolle an eine CloudWatch Logs-Protokollgruppe senden kann:

```
{
  "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 CloudWatch Protokolle gesendete Protokolle](#).

Amazon Data Firehose

- Sie geben einen Firehose-Bereitstellungs-Stream an, an den die Protokolle übermittelt werden.

- Protokolle aus mehreren Redis-Clustern und -Replikationsgruppen können an denselben Bereitstellungsdatenstrom 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 Protokollzustellung an Firehose ist derzeit in der Region Asien-Pazifik (Osaka) nicht verfügbar.

Berechtigungen zum Veröffentlichen von Protokollen in Firehose

Sie müssen über die folgenden Berechtigungen verfügen, damit ElastiCache Redis Protokolle an einen Amazon Kinesis Data Firehose-Bereitstellungs-Stream senden kann.

```
{
  "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

Mithilfe von AWS Management Console können Sie einen Redis-Cluster (Cluster-Modus deaktiviert) erstellen, indem Sie die Schritte unter [Einen Redis-Cluster erstellen \(Cluster-Modus deaktiviert\) \(Konsole\)](#) ausführen, oder Sie können einen Redis-Cluster (Cluster-Modus aktiviert) mithilfe der Schritte unter [Erstellen eines Redis-Clusters \(Cluster-Modus aktiviert\) \(Konsole\)](#) erstellen. In beiden Fällen konfigurieren Sie die Protokollzustellung wie folgt:

1. Wählen Sie unter Advanced Redis settings (Erweiterte Redis-Einstellungen) Logs (Protokolle) aus und aktivieren Sie dann entweder Slow logs (Langsame Protokolle) oder Engine logs (Engine-Protokolle).
2. Unter Protokollformat wählen Sie entweder Text oder JSON aus.
3. Unter Zieltyp wählen Sie entweder CloudWatch Logs oder Kinesis Firehose aus.
4. Wählen Sie unter Log destination (Protokollziel) entweder Create new (Neu erstellen) aus und geben Sie dann Ihren Amazon-S3-Bucket-Namen, CloudWatchLogs-Protokollgruppennamen oder Ihren Kinesis-Data-Firehose-Datenstromnamen aus oder wählen Sie Select existing (Vorhandene auswählen) aus und geben Sie einen der beiden letzten Namen ein.

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 ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis clusters (Redis-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:
 - Unter Zieltyp wählen Sie entweder CloudWatch Logs oder Kinesis Firehose aus.
 - Wählen Sie unter Protokollziel entweder Neu erstellen und geben Sie entweder den Namen Ihrer CloudWatchLogs-Protokollgruppe oder den Namen Ihres Kinesis-Data-Firehose-Streams ein. Oder Sie wählen Vorhandene auswählen und wählen dann entweder den Namen Ihrer CloudWatchLogs-Protokollgruppe oder den Namen Ihres Kinesis-Data-Firehose-Streams.

Angeben der Protokollzustellung mithilfe der AWS CLI

Slow-Protokoll

Erstellen Sie eine Replikationsgruppe mit langsamer Protokollzustellung an - CloudWatch Protokolle.

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 einer Replikationsgruppe, um langsame Protokolle an - CloudWatch Protokolle zu übermitteln

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 Engine-Protokollzustellung an - CloudWatch Protokolle.

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 einer 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 mithilfe von CloudWatch-Metriken

ElastiCache bietet Metriken, mit denen Sie Ihre Cluster überwachen können. Sie können über CloudWatch auf diese Metriken zugreifen. Weitere Informationen zu CloudWatch finden Sie in der [CloudWatch-Dokumentation](#).

ElastiCache bietet sowohl Metriken auf Host-Ebene (z. B. CPU-Auslastung) als auch für die Cache-Engine-Software spezifische Metriken (z. B. Cache-Gets und Cache-Misses). Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.

Important

Sie sollten erwägen, CloudWatch-Alarme für bestimmte Schlüsselmetriken festzulegen, damit Sie benachrichtigt werden, wenn die Leistung Ihres Cache-Clusters nachlässt. Weitere Informationen finden Sie unter [Welche Metriken sollte ich überwachen?](#) in diesem Handbuch.

Themen

- [Metriken auf Host-Ebene](#)
- [Metriken für Redis](#)
- [Welche Metriken sollte ich überwachen?](#)
- [Auswählen von Metrikstatistiken und -zeiträumen](#)
- [Überwachung von CloudWatch-Cluster- und Knoten-Metriken](#)

Metriken auf Host-Ebene

Der AWS/ElastiCache-Namespaces 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 Redis](#)

| Metrik | Beschreibung | Einheit |
|------------------|--|-------------------------|
| CPUUtilization | <p>Der Prozentsatz der CPU-Nutzung für den gesamten Host. Da Redis nur über einen Thread verfügt, empfehlen wir, die Metrik <code>EngineCPUUtilization</code> auf Knoten mit 4 oder mehr vCPUs zu überwachen.</p> | Prozent |
| CPUCreditBalance | <p>Die Anzahl verdienter CPU-Guthaben, die eine Instance angesammelt hat, seit sie gestartet wurde. Für T2 Standard beinhaltet der <code>CPUCreditBalance</code> auch die Anzahl der angesammelten Startguthaben.</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 Guthaben in der <code>CPUCreditBalance</code> sind verfügbar, um die Leistung der Instance über die Baseline ihrer CPU-Nutzung hinaus zu steigern.</p> <p>Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz verfügbar.</p> <p>Diese Metrik ist nicht für T2-Instances mit Spitzenleistung verfügbar.</p> | Guthaben (vCPU-Minuten) |

| Metrik | Beschreibung | Einheit |
|-----------------|---|-------------------------|
| CPUCreditUsage | <p>Die Anzahl der von der Instance für die CPU-Nutzung verbrauchten CPU-Guthaben. Ein CPU-Guthaben entspricht einer vCPU mit einer Auslastung von 100 % und einer Nutzungsdauer von einer Minute oder einer äquivalenten Kombination von vCPUs, Auslastung und Nutzungsdauer (z. B. eine vCPU mit einer Auslastung von 50 % mit einer Nutzungsdauer von zwei Minuten oder zwei vCPUs mit einer Auslastung von 25 % und einer Nutzungsdauer von zwei Minuten).</p> <p>Die Metriken für CPU-Guthaben sind nur mit einer fünfminütigen Frequenz 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> | Guthaben (vCPU-Minuten) |
| FreeableMemory | Größe des freien Arbeitsspeichers auf dem Host. Dies ist von RAM, Puffern und Cache abgeleitet, die vom Betriebssystem als freigebbar eingestuft 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 |

| Metrik | Beschreibung | Einheit |
|--|--|---------|
| 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 |
| 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 Steigerung der empfangenen Bytes innerhalb jeder Minute. | Bytes |

| Metrik | Beschreibung | Einheit |
|----------------------|---|---------|
| NetworkMaxBytesOut | Die maximale Steigerung der übertragenen Bytes innerhalb jeder Minute. | Bytes |
| NetworkMaxPacketsIn | Die maximale Steigerung der empfangenen Pakete innerhalb jeder Minute. | Anzahl |
| NetworkMaxPacketsOut | Die maximale Steigerung der übertragenen Pakete innerhalb jeder Minute. | Anzahl |
| SwapUsage | Größe des belegten Auslagerungsspeichers auf dem Host. | Bytes |

Metriken für Redis

Der AWS/ElastiCache-Namespace enthält die folgenden Redis-Metriken.

Mit Ausnahme von `ReplicationLag` und `EngineCPUUtilization` werden diese Metriken vom Redis-Befehl `info` abgeleitet. Jede Metrik wird zu jeder Cache-Knotenebene berechnet.

Die vollständige Dokumentation für den Redis-Befehl `info` finden Sie unter <http://redis.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 ist von der Statistik <code>active_defrag_hits</code> unter Redis INFO abgeleitet. | Zahl |
| AuthenticationFailures | Die Gesamtzahl der fehlgeschlagenen Versuche, sich mit dem AUTH-Befehl bei Redis | Anzahl |

| Metrik | Beschreibung | Einheit |
|--------------------|--|---------|
| | zu authentifizieren. Weitere Informationen zu einzelnen Authentifizierungsfehlern finden Sie mit dem Befehl ACL LOG . Wir empfehlen , hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen. | |
| | Die Gesamtzahl der von Redis für alle Zwecke zugewiesenen Bytes, einschließlich Datensatz, Puffer usw. | Bytes |
| BytesUsedForCache | Dimension: Tier=Memory für Redis-Cluster mit Daten-Tiering : Gesamtzahl der Bytes, die vom Speicher für den Cache verwendet werden. Das ist der Wert der <code>used_memory</code> - Statistik bei Redis INFO . | Bytes |
| | Dimension: Tier=SSD für Redis-Cluster mit Daten-Tiering : Gesamtzahl der Bytes, die vom SSD für den Cache verwendet werden. | Bytes |
| BytesReadFromDisk | Gesamtzahl der von der Festplatte pro Minute gelesenen Bytes. Wird nur für Cluster mit Daten-Tiering unterstützt. | Bytes |
| BytesWrittenToDisk | Gesamtzahl der Bytes, die pro Minute auf den Datenträger geschrieben werden. Wird nur für Cluster mit Daten-Tiering unterstützt. | Bytes |
| CacheHits | Die Anzahl der erfolgreichen schreibgeschützten Schlüsselsuchereignisse im Hauptverzeichnis. Dies ist von der Statistik <code>keyspace_hits</code> unter Redis INFO abgeleitet. | Anzahl |


| Metrik | Beschreibung | Einheit |
|------------------------------|--|---------|
| CacheMisses | Die Anzahl der nicht erfolgreichen schreibgeschützten Schlüsselsuchereignisse im Hauptverzeichnis. Dies ist von der Statistik <code>keyspace_misses</code> unter Redis INFO abgeleitet. | 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 Befehl ACL LOG . Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen. | Anzahl |
| CacheHitRate | Gibt die Nutzungseffizienz der Redis-Instanz an. Wenn das Cache-Verhältnis unter ~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: $\frac{\text{cache_hits}}{\text{cache_hits} + \text{cache_misses}}$ | 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 Befehl ACL LOG . 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 ist von <code>connected_clients</code> unter Redis INFO abgeleitet. | Anzahl |
| <code>CurrItems</code> | Anzahl der Elemente im Cache. Dies wird aus der Redis-Statistik <code>keyspace</code> abgeleitet, indem alle Schlüssel im gesamten Schlüsselbereich summiert werden. | Anzahl |
| | Dimension: <code>Tier=Memory</code> für Cluster mit Daten-Tiering . Anzahl der Elemente im Speicher. | Anzahl |
| | Dimension: <code>Tier=SSD</code> (Solid-State-Laufwerke) für Cluster mit Daten-Tiering . Anzahl der Elemente im SSD. | Anzahl |
| <code>CurrVolatileItems</code> | Gesamtzahl der Schlüssel in allen Datenbanken mit festgelegtem TTL. Dies wird aus der Redis-Statistik <code>expires</code> abgeleitet, indem alle Schlüssel mit festgelegtem TTL im gesamten Schlüsselbereich 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 wie $(\text{used_memory} - \text{mem_not_counted_for_evict} + \text{SSD used}) / (\text{maxmemory} + \text{SSD total capacity})$ folgt berechnet <code>used_memory</code> und <code>maxmemory</code> stammt aus Redis INFO.</p> <p>In allen anderen Fällen wird die Metrik mit $\text{used_memory}/\text{maxmemory}$ berechnet.</p> | Prozent |
| DatabaseCapacityUsageCountedForEvictPercentage | <p>Prozentsatz der gesamten Datenkapazität für den Cluster, die genutzt wird, ausschließlich des Speichers, der für Overhead und COB verwendet wird. 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>wobei <code>used_memory</code> und <code>maxmemory</code> aus Redis INFO stammen</p> | Prozent |
| DatabaseMemoryUsagePercentage | <p>Prozentsatz des Speichers für den Cluster, der genutzt wird. Eine Berechnung erfolgt mit $\text{used_memory}/\text{maxmemory}$ aus Redis INFO.</p> | Prozent |

| Metrik | Beschreibung | Einheit |
|--|---|---------------|
| DatabaseMemoryUsageCountedForEvictPercentage | Prozentsatz des Speichers für den Cluster, der genutzt wird, ausschließlich des Speichers, der für Overhead und COB verwendet wird.. Eine Berechnung erfolgt mit <code>used_memory-mem_not_counted_for_evict/maxmemory</code> aus Redis INFO . | Prozent |
| DB0AverageTTL | Stellt <code>avg_ttl</code> von DBO aus der keyspace-Statistik des Redis INFO -Befehls dar. Bei Replikaten laufen die Schlüssel nicht ab, die Replikate warten, bis die Schlüssel auf den Primärknoten ablaufen. Wenn auf einem Primärknoten ein Schlüssel abläuft (oder der Schlüssel wegen LRU bereinigt wird), synthetisiert der Primärschlüssel einen DEL-Befehl, der an alle Replikate übertragen wird. Daher lautet DB0AverageTTL für Replikatknoten 0, da dort keine Schlüssel ablaufen und somit TTL nicht verfolgt wird. | Millisekunden |


| Metrik | Beschreibung | Einheit |
|----------------------|---|---------|
| EngineCPUUtilization | <p>Gibt die CPU-Auslastung des Redis-Engine-Threads an. Weil Redis nur über einen Thread verfügt, können Sie diese Metrik zum Analysieren der Auslastung des Redis-Prozesses selbst verwenden. Die EngineCPUUtilization-Metrik bietet eine genauere Sichtbarkeit des Redis-Prozesses. Sie können dies in Verbindung mit der CPUUtilization-Metrik verwenden. CPUUtilization legt die CPU-Auslastung für die Server-Instance als Ganzes offen, einschließlich anderer Betriebssystem- und Verwaltungsprozesse. Verwenden Sie für größere Knotentypen mit vier vCPUs oder mehr die EngineCPUUtilization-Metrik zum Überwachen und Einstellen der Schwellenwerte für die Skalierung.</p> | Prozent |

 Note


Auf einem ElastiCache Host überwachen Hintergrundprozesse den Host, um eine verwaltete Datenbankumgebung bereitzustellen. Diese Hintergrundprozesse können einen erheblichen Teil der CPU-Workload beanspruchen. Dies ist auf größeren Hosts mit mehr als zwei vCPUs nicht signifikant. Es kann sich jedoch auf kleinere Hosts mit 2vCPUs oder weniger auswirken. Wenn Sie nur die EngineCPUUtilization-Metrik überwachen, werden Sie keine Situationen erkennen, in denen der Host sowohl mit hoher CPU-Auslastung von Redis als auch mit hoher CPU-Auslastung aus den

| Metrik | Beschreibung | Einheit |
|-------------------------------|--|----------|
| | Hintergrundüberwachungsprozessen überlastet ist. Daher empfehlen wir die Überwachung der CPUUtilization - Metrik für Hosts mit zwei vCPUs oder weniger. | |
| Evictions | Anzahl der Schlüssel, die infolge des maxmemory -Grenzwertes bereinigt worden sind. Dies ist von evicted_keys unter Redis INFO abgeleitet. | Anzahl |
| GlobalDatastoreReplicationLag | Dies ist die Verzögerung zwischen dem Primärknoten der sekundären Region und dem Primärknoten der primären Region. Bei redis mit aktiviertem Cluster-Modus, zeigt die Verzögerung die maximale Verzögerung zwischen den Shards an. | Sekunden |
| IamAuthenticationExpirations | Die Gesamtzahl der abgelaufenen IAM-authentifizierten Redis-Verbindungen. Weitere Informationen über Authentifizieren mit IAM finden Sie im Benutzerhandbuch. | Anzahl |
| IamAuthenticationThrottling | Die Gesamtzahl der gedrosselten IAM-authentifizierten Redis AUTH- oder HELLO-Anfragen. Weitere Informationen über Authentifizieren mit IAM 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 ACL LOG . Wir empfehlen, hierauf einen Alarm zu setzen, um unberechtigte Zugriffsversuche zu erkennen. | Anzahl |
| KeysTracked | Die Anzahl der Schlüssel, die von der Redis-Schlüsselverfolgung verfolgt werden, in Prozent 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 Redis Engine an. Bestimmte Schwellenwerte weisen auf unterschiedliche Verhaltensweisen hin. Der empfohlene Wert ist eine Fragmentierung über 1,0. Diese wird aus der <code>mem_fragmentation_ratio</code> statistic von Redis INFO berechnet. | Zahl |

| Metrik | Beschreibung | Einheit |
|------------------------|---|----------|
| NewConnections | <p>Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind. Dies ist von <code>total_connections_received</code> unter Redis INFO abgeleitet.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Wenn Sie ElastiCache für Redis Version 5 oder niedriger verwenden, werden zwischen zwei und vier der von dieser Metrik gemeldeten Verbindungen ElastiCache zur Überwachung des Clusters verwendet. ElastiCache Bei Verwendung von Redis Version 6 oder höher sind die Verbindungen, die von ElastiCache zur Überwachung des Clusters verwendet werden, jedoch nicht in dieser Metrik enthalten.</p> </div> | Anzahl |
| NumItemsReadFromDisk | Die Gesamtzahl der pro Minute von der Festplatte abgerufenen Elemente. Wird nur für Cluster mit Daten-Tiering unterstützt. | Anzahl |
| NumItemsWrittenToDisk | Die Gesamtzahl der pro Minute auf die Festplatte geschriebenen Elemente. Wird nur für Cluster mit Daten-Tiering unterstützt. | Anzahl |
| MasterLinkHealthStatus | Dieser Status kann zwei Werte annehmen: 0 oder 1. Der Wert 0 gibt an, dass die Daten im ElastiCache Primärknoten nicht mit Redis auf EC2 synchronisiert sind. Der Wert 1 bedeutet, dass die Daten synchronisiert sind. Verwenden Sie den CompleteMigration API-Vorgang, um die Migration abzuschließen. | Boolesch |

| Metrik | Beschreibung | Einheit |
|------------------|--|----------|
| Reclaimed | Gesamtanzahl der Schlüsselablaufereignisse Dies ist von <code>expired_keys</code> unter Redis INFO abgeleitet. | 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 ist von <code>master_repl_offset</code> unter Redis INFO abgeleitet. | 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. Ab Redis-Engine-Version 5.0.6 kann die Verzögerung in Millisekunden gemessen werden. | Sekunden |
| 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 <code>SaveInProgress</code> - Metrik können Sie diagnostizieren, ob die Leistungsbeeinträchtigung von einem Hintergrundspeichervorgang verursacht wurde. Dies ist von <code>rdb_bgsave_in_progress</code> unter Redis INFO abgeleitet. | Boolesch |

| Metrik | Beschreibung | Einheit |
|-------------------------|---|----------|
| TrafficManagementActive | <p>Gibt an, ob ElastiCache for Redis den Datenverkehr aktiv verwaltet, indem der den eingehenden Befehlen zugewiesene Verkehr angepasst, überwacht oder repliziert wird. Der Datenverkehr wird verwaltet, wenn mehr Befehle an den Knoten gesendet werden, als von Redis verarbeitet werden können, und dies wird verwendet, um die Stabilität und den optimalen Betrieb der Engine aufrechtzuerhalten. 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> Note</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 |

Verfügbarkeit von EngineCPUUtilization

AWS Die unten aufgeführten Regionen sind auf allen 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`“ (Befehlsstatistiken) bietet Statistiken auf der Grundlage des Befehlstyps, einschließlich der Anzahl der Aufrufe, des gesamten durch diese Befehle verursachten CPU-Zeitaufwands und des durchschnittlichen CPU-Verbrauchs pro Befehlsausführung. Für jeden Befehlstyp wird die folgende Zeile hinzugefügt: `cmdstat_XXX: calls=XXX,usec=XXX,usec_per_call=XXX`.

Die folgend aufgeführten Latenzmetriken werden mithilfe der `commandstats`-Statistik von [Redis INFO](#) berechnet. Diese werden auf folgende Weise berechnet: $\text{delta}(\text{usec})/\text{delta}(\text{calls})$. `delta` wird als Differenz innerhalb einer Minute berechnet. Latenz ist definiert als CPU-Zeit, die ElastiCache für die Verarbeitung des Befehls benötigt wird. Beachten Sie, dass für Cluster, die Daten-Tiering verwenden, die zum Abrufen von Elementen vom SSD benötigte Zeit in diesen Messungen nicht enthalten ist.

Eine vollständige Liste der verfügbaren Befehle finden Sie unter [Redis-Befehle](#) in der Redis-Dokumentation.

| Metrik | Beschreibung | Einheit |
|--------------------------------------|--|---------------|
| <code>ClusterBasedCmds</code> | Die Gesamtanzahl der Cluster-basierten Befehle. Diese wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle Befehle, die auf einen oder mehrere Cluster (<code>cluster slot</code> , <code>cluster info</code> usw.) wirken, summiert werden. | Anzahl |
| <code>ClusterBasedCmdsLatency</code> | Latenz von Cluster-basierten Befehlen. | Mikrosekunden |

| Metrik | Beschreibung | Einheit |
|----------------------------|---|---------------|
| EvalBasedCmds | Die Gesamtzahl der Befehle für EVAL-basierte Befehle. Diese wird aus der Redis-commandstats -Statistik durch Summieren von eval, evalsha abgeleitet. | Anzahl |
| EvalBasedCmdsLatency | Latenz von eval-basierten Befehlen. | Mikrosekunden |
| GeoSpatialBasedCmds | Die Gesamtzahl der Befehle für raumbezogene Befehle. Diese wird aus der Statistik der Redis-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 Redis-Statistik commandstats abgeleitet, in der alle read-only-Befehle (get, hget, scard, lrange usw.) summiert werden | Anzahl |
| GetTypeCmdsLatency | Latenz von Lesebefehlen. | Mikrosekunden |
| HashBasedCmds | Gesamtanzahl der Hash-basierten Befehle. Diese wird aus der Redis-Statistik commandstats abgeleitet, indem alle Befehle, die auf eine oder mehrere Hashes wirken, summiert werden (hget, hkeys, hvals, hdel usw.). | Anzahl |
| HashBasedCmdsLatency | Latenz von Hash-basierten Befehlen. | Mikrosekunden |

| Metrik | Beschreibung | Einheit |
|-----------------------------|---|---------------|
| HyperLogLogBasedCmds | Gesamtanzahl der auf HyperLogLog basierenden Befehle. Dies wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle pf-Befehle (<code>pfadd</code> , <code>pfcount</code> , <code>pfmerge</code> usw.) summiert werden. | 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 von Redis abgeleitet, indem alle JSON-Befehle summiert werden, die auf JSON-Schlüssel einwirken. | Anzahl |
| JsonBasedCmdsLatency | Die Latenz der JSON-Befehle, einschließlich Lese- und Schreibbefehlen. | Mikrosekunden |
| JsonBasedGetCmds | Gesamtanzahl der JSON-Schreibschutzbefehle. Dies wird aus der <code>commandstats</code> -Statistik von Redis abgeleitet, indem alle JSON-Lesebefehle summiert werden, die auf JSON-Schlüssel einwirken. | Anzahl |
| JsonBasedGetCmdsLatenz | Latenz der JSON-Schreibschutzbefehle. | Mikrosekunden |
| JsonBasedSetCmds | Gesamtanzahl der JSON-Schreibbefehle. Dies wird aus der <code>commandstats</code> -Statistik von Redis abgeleitet, indem alle JSON-Schreibbefehle summiert werden, die auf JSON-Schlüssel einwirken. | Anzahl |
| JsonBasedSetCmdsLatenz | Latenz von JSON-Schreibbefehlen. | Mikrosekunden |

| Metrik | Beschreibung | Einheit |
|-----------------------|---|---------------|
| KeyBasedCmds | Gesamtanzahl der schlüsselbasierten Befehle. Dies wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle Befehle summiert werden, die über mehrere Datenstrukturen hinweg auf mindestens einen Schlüssel einwirken (<code>del</code> , <code>expire</code> , <code>rename</code> usw.). | Anzahl |
| KeyBasedCmdsLatency | Latenz von schlüsselbasierten Befehlen. | Mikrosekunden |
| ListBasedCmds | Gesamtanzahl der listenbasierten Befehle. Dieser Wert wird aus der Redis-Statistik <code>commandstats</code> abgeleitet, indem alle Befehle summiert werden, die einzelne oder mehrere Listen bearbeiten (<code>lindex</code> , <code>lrange</code> , <code>lpush</code> , <code>ltrim</code> usw.). | Anzahl |
| ListBasedCmdsLatency | Latenz von listenbasierten Befehlen. | Mikrosekunden |
| NonKeyTypeCmds | Gesamtanzahl der nicht schlüsselbasierten Befehle. Dieser wird aus der <code>commandstats</code> -Statistik von Redis abgeleitet, indem alle Befehle summiert werden, die nicht auf einen Schlüssel einwirken, z. B.: <code>acl</code> , <code>dbsize</code> oder <code>info</code> . | Anzahl |
| NonKeyTypeCmdsLatency | Latenz der non-key-based Befehle. | Mikrosekunden |

| Metrik | Beschreibung | Einheit |
|------------------------|---|---------------|
| PubSubBasedCmds | Die Gesamtzahl der Befehle für die Pub/Sub-Funktionalität. Diese wird aus den Redis-commandstats -Statistiken abgeleitet, indem alle für die pub/sub-Funktionalität verwendeten Befehle addiert werden: psubscribe, publish, pubsub, punsubscribe, ssubscribe, sunsubscribe, spublish, subscribe und unsubscribe. | Anzahl |
| PubSubBasedCmdsLatency | Latenz von pub/sub-basierten Befehlen. | Mikrosekunden |
| SetBasedCmds | Gesamtanzahl der Set-basierten Befehle. Diese wird aus der Redis-Statistik commandstats abgeleitet, indem alle Befehle, die auf eine oder mehrere Sets wirken, summiert werden (scard, sdiff, sadd, sunion usw.). | Anzahl |
| SetBasedCmdsLatency | Latenz von Set-basierten Befehlen. | Mikrosekunden |
| SetTypeCmds | Gesamtanzahl der auf write basierenden Befehle. Dieser Wert wird von der Redis-commandstats -Statistik abgeleitet, indem alle Arten von mutative-Befehlen, die auf Daten wirken, addiert werden (set, hset, sadd, lpop usw.). | Anzahl |
| SetTypeCmdsLatency | Latenz von Schreibbefehlen. | Mikrosekunden |

| Metrik | Beschreibung | Einheit |
|---------------------------|---|---------------|
| SortedSetBasedCmds | Gesamtanzahl der Sorted Set-basierten Befehle. Sie wird von der Redis-commandstats -Statistik abgeleitet, indem alle Befehle, die auf eine oder mehrere sortierte Gruppen wirken, addiert werden (zcount, zrange, zrank, zadd usw.). | Anzahl |
| SortedSetBasedCmdsLatency | Latenz von Sortierungs-basierten Befehlen. | Mikrosekunden |
| StringBasedCmds | Gesamtanzahl der Zeichenfolge-basierten Befehle. Diese wird aus der Redis-commandstats -Statistik abgeleitet, indem alle Befehle, die auf eine oder mehrere Zeichenketten wirken, summiert werden (strlen, setex, setrange usw.). | Anzahl |
| StringBasedCmdsLatency | Latenz von Zeichenfolgen-basierten Befehlen. | Mikrosekunden |
| StreamBasedCmds | Die Gesamtanzahl Stream-basierter Befehle. Dieser Wert wird aus der Redis-commandstats -Statistik abgeleitet, indem alle Befehle summiert werden, die einzelne oder mehrere Stream-Datentypen bearbeiten (xrange, xlen, xadd, xdel usw.). | Anzahl |
| StreamBasedCmdsLatency | Latenz von Stream-basierten Befehlen. | Mikrosekunden |

Welche Metriken sollte ich überwachen?

Die folgenden CloudWatch-Metriken bieten hilfreiche Einblicke in die Leistung von ElastiCache. In den meisten Fällen empfehlen wir, für diese Metriken CloudWatch-Alarme einzurichten, damit Sie im Fall von Leistungsabfällen Korrekturmaßnahmen ergreifen können.

Zu überwachende Metriken

- [CPUUtilization](#)
- [EngineCPUUtilization](#)
- [SwapUsage](#)
- [Evictions](#)
- [CurrConnections](#)
- [Arbeitsspeicher](#)
- [Network \(Netzwerk\)](#)
- [Latency](#)
- [Replikation](#)
- [Verwaltung des Datenverkehrs](#)

CPUUtilization

Diese Metrik auf Hostebene wird in Prozent angegeben. Weitere Informationen finden Sie unter [Metriken auf Host-Ebene](#).

Verwenden Sie für kleinere Knotentypen mit 2vCPUs oder weniger die Metrik `CPUUtilization`, um Ihre Workload zu überwachen.

Generell empfehlen wir, den Schwellenwert auf 90 % der verfügbaren CPU-Kapazität festzulegen. Da Redis mit nur einem Thread arbeitet, sollte der tatsächliche Schwellenwert als ein Bruchteil der Gesamtkapazität des Knotens berechnet werden. Angenommen, Sie verwenden einen Knotentyp mit zwei Kernen. In diesem Fall wäre der Grenzwert für `CPUUtilization` $90/2$ bzw. 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:

- Redis-Cluster(Cluster-Modus deaktiviert): Die Skalierung erfolgt mithilfe eines größeren Cache-Instance-Typs.
- Redis-Cluster (Cluster-Modus aktiviert): Fügen Sie weitere Shards hinzu, um den Schreib-Workload auf mehr Primärknoten zu verteilen.

Tip

Anstatt die Metrik `CPUUtilization` auf Hostebene zu verwenden, können Redis-Benutzer möglicherweise die Redis-Metrik `EngineCPUUtilization` verwenden, die den Prozentsatz der Auslastung des Redis-Engine-Kerns angibt. Informationen dazu, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere hilfreiche Informationen finden Sie unter [Metriken für Redis](#).

Verwenden Sie für größere Knotentypen mit 4vCPUs oder mehr die Metrik `EngineCPUUtilization`, die den Prozentsatz der Auslastung des Redis-Engine-Kerns angibt. Informationen dazu, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere hilfreiche Informationen finden Sie unter [Metriken für Redis](#).

EngineCPUUtilization

Verwenden Sie für größere Knotentypen mit 4vCPUs oder mehr die Metrik `EngineCPUUtilization`, die den Prozentsatz der Auslastung des Redis-Engine-Kerns angibt. Informationen dazu, ob diese Metrik auf Ihren Knoten verfügbar ist, und weitere hilfreiche Informationen finden Sie unter [Metriken für Redis](#).

Weitere Informationen finden Sie im Abschnitt CPUs unter [Überwachung von bewährten Methoden mit Amazon ElastiCache for Redis unter Verwendung von Amazon CloudWatch](#).

SwapUsage

Diese Metrik auf Hostebene wird in Bytes angegeben. Weitere Informationen finden Sie unter [Metriken auf Host-Ebene](#).

Wenn die CloudWatch-Metrik `FreeableMemory` einen Wert nahe 0 aufweist (d. h. unter 100 MB) oder die Metrik `SwapUsage` größer als die Metrik `FreeableMemory` ist, weist dies darauf hin, dass ein Knoten unter Speicherdruck steht. Beachten Sie in diesem Fall folgende Themen

- [Stellen Sie sicher, dass Sie über genügend Arbeitsspeicher verfügen, um einen Redis-Snapshot zu erstellen](#)

- [Verwalten von reserviertem Speicher](#)

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.

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.

Ein höherer Wert für CurrConnections kann auf ein Problem mit der Anwendung hinweisen. Untersuchen Sie das Verhalten der Anwendung, um das Problem zu beheben.

Weitere Informationen finden Sie im Abschnitt Verbindungen unter [Überwachung von bewährten Methoden mit Amazon ElastiCache for Redis unter Verwendung von Amazon CloudWatch](#).

Arbeitsspeicher

Speicher ist ein zentraler Aspekt von Redis. Es ist notwendig, die Speicherauslastung Ihres Clusters zu verstehen, um Datenverluste zu vermeiden und das zukünftige Wachstum Ihres Datensets berücksichtigen zu können. Statistiken zur Speicherauslastung eines Knotens finden Sie im Speicherbereich des Redis [INFO](#)-Befehls.

Weitere Informationen finden Sie im Abschnitt Speicher unter [Überwachung von bewährten Methoden mit Amazon ElastiCache for Redis unter Verwendung von Amazon CloudWatch](#).

Network (Netzwerk)

Einer der entscheidenden Faktoren für die Kapazität der Netzwerkbandbreite Ihres Clusters ist der von Ihnen ausgewählte Knotentyp. Weitere Informationen zur Netzwerkkapazität Ihres Knotens finden Sie unter [Preisgestaltung für Amazon ElastiCache](#).

Weitere Informationen finden Sie im Abschnitt Netzwerk unter [Überwachung von bewährten Methoden mit Amazon ElastiCache for Redis unter Verwendung von Amazon CloudWatch](#).

Latency

Sie können die Latenz eines Befehls mit einer Reihe von CloudWatch-Metriken messen, die aggregierte Latenzen pro Datenstruktur bereitstellen. Die unten aufgeführten Latenzmetriken werden mithilfe der `commandstats`-Statistik vom Redis-[INFO](#)-Befehl berechnet.

Weitere Informationen finden Sie im Abschnitt Latenz unter [Überwachung von bewährten Methoden mit Amazon ElastiCache for Redis unter Verwendung 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 [Überwachung von bewährten Methoden mit Amazon ElastiCache for Redis unter Verwendung von Amazon CloudWatch](#).

Verwaltung des Datenverkehrs

ElastiCache für Redis verwaltet den Datenverkehr für einen Knoten automatisch, wenn mehr eingehende Befehle an den Knoten gesendet werden, als von Redis verarbeitet werden können. 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 unterkalkuliert 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

Sie können in CloudWatch zwar beliebige Statistiken und Zeiträume für einzelne Metriken festlegen, jedoch sind nicht alle Kombinationen sinnvoll. So sind zum Beispiel die Statistiken Durchschnitt, Minimum und Maximum für CPUUtilization nützlich, die Statistik Summe jedoch nicht.

Alle ElastiCache-Beispiele werden für einen Zeitraum von 60 Sekunden für jeden 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 Knoten-Metriken](#).

Überwachung von CloudWatch-Cluster- und Knoten-Metriken

ElastiCache und CloudWatch sind integriert, sodass Sie eine Vielzahl von Metriken sammeln können. Sie können diese Metriken mit CloudWatch überwachen.

Note

Für das folgende Beispiel werden die CloudWatch-Befehlszeilen-Tools benötigt. Weitere Informationen zu CloudWatch und zum Download der Entwicklertools finden Sie auf der [CloudWatch-Produktseite](#).

Die folgenden Verfahren zeigen, wie Sie CloudWatch zum Erfassen von Speicherplatz-Statistiken für einen Cache-Cluster innerhalb der letzten Stunde verwenden.

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-Limits finden Sie unter [AWS-Service-Limits](#) für ElastiCache.

Überwachung von CloudWatch-Cluster- und Knoten-Metriken (Konsole)

So erfassen Sie CPU-Nutzungsstatistiken für einen Cache-Cluster

1. Melden Sie sich bei AWS Management Console an 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-Managementkonsole 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.

Unten im Konsolenfenster erscheint eine Liste verfügbarer CloudWatch-Metriken.

- d. Klicken Sie auf die Metrik CPU Utilization.

Die CloudWatch-Konsole wird geöffnet und zeigt die 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 Knoten-Metriken mithilfe des CloudWatch-CLI

So erfassen Sie CPU-Nutzungsstatistiken für einen Cache-Cluster

- 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"}]' \  
  --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 Knoten-Metriken mithilfe der CloudWatch-API

So erfassen Sie CPU-Nutzungsstatistiken für einen Cache-Cluster

- Rufen Sie die CloudWatch-API `GetMetricStatistics` mit den folgenden Parametern auf (beachten Sie, dass Start- und Endzeitpunkt nur als Beispiele dienen; Sie müssen Ihren eigenen Start- und Endzeitpunkt einsetzen):
 - `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>
```

Amazon-SNS-Überwachung von ElastiCache-Ereignissen

Wenn es auf einem Cluster zu wichtigen Ereignissen kommt, sendet ElastiCache Benachrichtigungen an ein bestimmtes Amazon-SNS-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

- [Verwalten von ElastiCache-Amazon-SNS-Benachrichtigungen](#)
- [Anzeigen von ElastiCache-Ereignissen](#)
- [Ereignisbenachrichtigungen und Amazon SNS](#)

Verwalten von ElastiCache-Amazon-SNS-Benachrichtigungen

Sie können ElastiCache so konfigurieren, dass Benachrichtigungen für wichtige Cluster-Ereignisse mit Amazon Simple Notification Service (Amazon SNS) gesendet werden. In diesen Beispielen konfigurieren Sie einen Cluster mit dem Amazon-Ressourcenname (ARN) eines Amazon-SNS-Themas, um Benachrichtigungen zu erhalten.

Note

In diesem Thema wird davon ausgegangen, dass Sie sich bei Amazon SNS angemeldet und ein Amazon-SNS-Thema eingerichtet und abonniert haben. Informationen dazu finden Sie im [Entwicklerhandbuch zu Amazon Simple Notification Service](#).

Hinzufügen eines Amazon-SNS-Themas

In den folgenden Abschnitten erfahren Sie, wie Sie ein Amazon-SNS-Thema mithilfe der AWS-Konsole, dem AWS CLI oder der ElastiCache-API hinzufügen.

Hinzufügen eines Amazon-SNS-Themas (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie ein Amazon-SNS-Thema für einen Cluster hinzufügen. Um ein Amazon-SNS-Thema für eine Replikationsgruppe hinzuzufügen, wählen Sie in Schritt 2 anstelle eines Clusters eine Replikationsgruppe aus und befolgen Sie die übrigen Schritte.

Note

Diese Vorgehensweise kann auch zum Ändern des Amazon-SNS-Themas verwendet werden.

So fügen Sie ein Amazon-SNS-Thema für einen Cluster hinzu oder ändern es (Konsole)

1. Melden Sie sich bei AWS Management Console an 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 Amazon-SNS-Thema-ARN hinzufügen oder ändern möchten.
3. Wählen Sie Ändern aus.
4. Wählen Sie im Feld Modify Cluster (Cluster ändern) unter Topic for SNS Notification (Thema für SNS-Benachrichtigung) das SNS-Thema aus, das Sie hinzufügen möchten, oder wählen Sie Manual ARN input (Manuelle ARN-Eingabe) aus und geben Sie den ARN des Amazon-SNS-Themas ein.
5. Wählen Sie Ändern aus.

Hinzufügen eines Amazon-SNS-Themas (AWS CLI)

Verwenden Sie zum Hinzufügen oder Ändern eines Amazon-SNS-Themas für einen Cluster den AWS CLI-Befehl `modify-cache-cluster`.

Das folgende Codebeispiel fügt einen Amazon-SNS-Themen-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](#).

Hinzufügen eines Amazon-SNS-Themas (ElastiCache-API)

Rufen Sie zum Hinzufügen oder Ändern eines Amazon-SNS-Themas für einen Cluster die Aktion `ModifyCacheCluster` mit 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](#).

Aktivieren und Deaktivieren von Amazon-SNS-Benachrichtigungen

Sie können Benachrichtigungen für einen Cluster aktivieren oder deaktivieren. Das folgende Verfahren zeigt, wie Sie Amazon-SNS-Benachrichtigungen deaktivieren.

Aktivieren und Deaktivieren von Amazon-SNS-Benachrichtigungen (Konsole)

So deaktivieren Sie Amazon SNS-Benachrichtigungen mit der AWS Management Console

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste der Cluster anzuzeigen, die Redis ausführen, wählen Sie im Navigationsbereich Redis.
3. Aktivieren Sie das Kontrollkästchen links neben dem Cluster, für den die Benachrichtigung geändert werden soll.
4. Wählen Sie Ändern aus.
5. Wählen Sie im Feld Modify Cluster unter Topic for SNS Notification die Option Disable Notifications aus.
6. Wählen Sie Ändern aus.

Aktivieren und Deaktivieren von Amazon-SNS-Benachrichtigungen (AWS CLI)

Verwenden Sie zum Deaktivieren von Amazon-SNS-Benachrichtigungen den Befehl `modify-cache-cluster` mit folgenden Parametern:

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
```

Aktivieren und Deaktivieren von Amazon-SNS-Benachrichtigungen (ElastiCache-API)

Rufen Sie zum Deaktivieren von Amazon-SNS-Benachrichtigungen die `ModifyCacheCluster`-Aktion mit 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>
```

Anzeigen von ElastiCache-Ereignissen

ElastiCache protokolliert Ereignisse, die sich auf Ihre Cluster-Instances, Sicherheits- und Parametergruppen beziehen. Diese Informationen beinhalten Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mit der ElastiCache-Konsole, dem AWS CLI-Befehl `describe-events` oder der ElastiCache-API-Aktion `DescribeEvents` einfach aus dem Protokoll abrufen.

Die folgenden Schritte zeigen, wie Sie alle ElastiCache-Ereignisse der letzten 24 Stunden (1 440 Minuten) anzeigen.

Anzeigen von ElastiCache-Ereignissen (Konsole)

Das folgende Verfahren zeigt Ereignisse mithilfe der ElastiCache-Konsole an.

So zeigen Sie Ereignisse mithilfe der ElastiCache-Konsole an

1. Melden Sie sich bei AWS Management Console an 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).

Im Bildschirm Events (Ereignisse) repräsentiert jede Zeile in der Liste ein Ereignis und zeigt die Ereignisquelle, den Ereignistyp (`cache-cluster`, `cache-parameter-group`, `cache-security-group`, `cache-subnet-group` oder `replication-group`), die Uhrzeit (GMT) 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.

Anzeigen von ElastiCache-Ereignissen (AWS CLI)

Verwenden Sie zum Generieren einer Liste der ElastiCache-Ereignisse mithilfe des AWS CLI 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](#).

Anzeigen von ElastiCache-Ereignissen (ElastiCache-API)

Verwenden Sie zum Generieren einer Liste der ElastiCache-Ereignisse mithilfe der ElastiCache-API die Aktion `DescribeEvents`. 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](#).

Ereignisbenachrichtigungen 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 zum Amazon Simple Notification Service (SNS) sowie Informationen zu Preisen und Links zur Amazon-SNS-Dokumentation finden Sie auf der [Produktseite zu Amazon SNS](#).

Benachrichtigungen werden in einem bestimmten Amazon-SNS-Thema veröffentlicht. Für Benachrichtigungen müssen folgende Voraussetzungen erfüllt sein:

- Für ElastiCache-Benachrichtigungen kann nur ein Thema konfiguriert werden.
- Das AWS-Konto, das das Amazon-SNS-Thema besitzt, muss dasselbe Konto sein, das den Cache-Cluster besitzt, für den Benachrichtigungen aktiviert sind.
- Das Amazon-SNS-Thema, in dem Sie veröffentlichen, kann nicht verschlüsselt werden.

Note

Es ist möglich, ein verschlüsseltes (im Ruhezustand befindliches) Amazon-SNS-Thema an den Cluster anzuhängen. Der Status des Themas aus der ElastiCache-Konsole wird jedoch als inaktiv angezeigt, wodurch das Thema effektiv vom Cluster getrennt wird, wenn ElastiCache Nachrichten an das Thema sendet.

- Das Amazon-SNS-Thema muss sich in der gleichen Region wie der ElastiCache-Cluster befinden.


ElastiCache-Ereignisse


Die folgenden ElastiCache-Ereignisse lösen Amazon-SNS-Benachrichtigungen aus. Weitere Informationen zu Ereignisdetails finden Sie unter [Anzeigen von ElastiCache-Ereignissen](#).

| 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 wegen nicht genügend 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:CacheClusterProvisioningFailed : <i>cluster-name</i> | Es wurde versucht, einen neuen Cache-Cluster in einer nicht vorhandenen Virtual Private 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">• Die Liste der Cache-Sicherheitsgruppen, die für den Cache-Cluster autorisiert sind, wurde geändert.• Mindestens eine neue EC2-Sicherheitsgruppe wurde für eine der Cache-Sicherheitsgruppen autorisiert, die dem Cache-Cluster zugeordnet sind.• Mindestens eine EC2-Sicherheitsgruppe wurde aus den Cache-Sicherheitsgruppen entfernt, die dem Cache-Cluster zugeordnet sind. |

| Ereignisname | Fehlermeldung | Beschreibung |
|---|--|---|
| Elasticache:CacheNoderePlac eStarted | ElastiCache:CacheN odeReplaceStarted : <i>cluster-name</i> | <p>ElastiCache hat erkannt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, schwach oder nicht erreichbar ist, und hat damit 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> Note</p><p>Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert.</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 nicht mehr, selbst nachdem ElastiCache den Cache-Knoten ersetzt hat. 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 erkannt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, schwach oder nicht erreichbar ist, und hat den Cache-Knoten ersetzt.</p> <div data-bbox="1068 541 1507 856"><p> Note</p><p>Der DNS-Eintrag für den ersetzten Cache-Knoten bleibt unverändert.</p></div> <p>In den meisten Fällen müssen Sie die Serverliste für die Clients nicht aktualisieren, wenn dieses Ereignis auftritt. Einige Cache-Clients-Bibliotheken verwenden den Cache-Knoten jedoch möglicherweise nicht mehr, selbst nachdem ElastiCache den Cache-Knoten ersetzt hat. 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 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 Redis) | 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 Ersetzen von Knoten.</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 Ersetzen von Knoten.</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 Redis) | ElastiCache:SnapshotComplete : <i>cluster-name</i> | Ein Cache-Snapshot wurde erfolgreich abgeschlossen. |
| ElastiCache:SnapshotFailed (nur Redis) | 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 DescribeSnapshots), ist dessen Status failed. |

Verwandte Themen

- [Anzeigen von ElastiCache-Ereignissen](#)

Protokollieren von Amazon-ElastiCache-API-Aufrufen mithilfe von AWS CloudTrail

Amazon ElastiCache ist in AWS CloudTrail integriert. Dieser Service zeichnet die Aktionen eines Benutzers, einer Rolle oder eines AWS-Services in Amazon ElastiCache auf. CloudTrail erfasst alle API-Aufrufe für Amazon ElastiCache als Ereignisse, einschließlich Aufrufen von der Amazon-ElastiCache-Konsole und von Code-Aufrufen an die Amazon-ElastiCache-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon-S3-Bucket, einschließlich Ereignissen für Amazon ElastiCache, aktivieren. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse dennoch in der CloudTrail-Konsole im Ereignisverlauf anzeigen. Mit den von CloudTrail gesammelten Informationen können Sie die an Amazon ElastiCache gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Angaben bestimmen.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail-Benutzerhandbuch](#).

Informationen zu Amazon ElastiCache in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Kontos für Sie aktiviert. Wenn eine Aktivität in Amazon ElastiCache auftritt, wird diese Aktivität in einem CloudTrail-Ereignis zusammen mit anderen AWS-Service-Ereignissen im Ereignisverlauf aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail-Ereignisverlauf](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, darunter Ereignisse für Amazon ElastiCache, können Sie einen Pfad (Trail) erstellen. Ein Trail ermöglicht es CloudTrail, Protokolldateien in einem Amazon-S3-Bucket bereitzustellen. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser standardmäßig für alle Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- [Übersicht zum Erstellen eines Trails](#)
- [Von CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfigurieren von Amazon-SNS-Benachrichtigungen für CloudTrail](#)

- [Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail-Protokolldateien aus mehreren Konten](#)

Alle Amazon-ElastiCache-Aktionen werden von CloudTrail protokolliert und sind in der [API-Referenz von ElastiCache](#) dokumentiert. Zum Beispiel generieren Aufrufe der Aktionen `CreateCacheCluster`, `DescribeCacheCluster` und `ModifyCacheCluster` Einträge in den CloudTrail-Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Gibt an, ob die Anfrage mit Root- oder IAM-Benutzer-Anmeldeinformationen von ausgeführt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer gesendet wurde.
- Gibt an, ob die Anforderung aus einem anderen AWS-Service gesendet wurde

Weitere Informationen finden Sie unter dem [CloudTrail userIdentity-Element](#).

Grundlegendes zu den Amazon-ElastiCache-Einträgen in der Protokolldatei

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den von Ihnen angegebenen Amazon-S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolleinträge sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Das folgende Beispiel zeigt einen CloudTrail-Protokolleintrag, der die Aktion `CreateCacheCluster` 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 Aktion `DescribeCacheCluster` demonstriert. Beachten Sie, dass für alle Amazon-ElastiCache-Describe-Aufrufe (Describe*), der Abschnitt `ResponseElements` entfernt wird und als `null` angegeben 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 erfasst.

```
{
  "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"
}
```

ElastiCache-Kontingente

Das AWS-Konto verfügt über Standardkontingente (früher als Limits bezeichnet) für jeden AWS-Service. 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 ElastiCache anzuzeigen, öffnen Sie die [Konsole für Service Quotas](#). Wählen Sie im Navigationsbereich AWS-Services und ElastiCache aus.

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](#).

Ihr AWS-Konto umfasst die folgenden Kontingente für ElastiCache.

| Ressource | Standard |
|---|----------|
| Serverless-Caches pro Region | 40 |
| Serverless-Snapshots pro Tag und Cache | 24 |
| Knoten pro Region | 300 |
| Knoten pro Cluster pro Instance-Typ (Redis-Cluster-Modus aktiviert) | 90 |
| Knoten pro Shard (Redis-Cluster-Modus deaktiviert) | 6 |
| Parametergruppen pro Region | 300 |
| Sicherheitsgruppen pro Region | 50 |
| Subnetzgruppen pro Region | 300 |
| Subnetze pro Subnetzgruppe | 20 |
| Benutzer pro Benutzergruppe | 100 |
| Maximale Anzahl der Benutzer. | 1000 |

| Ressource | Standard |
|-------------------------------------|----------|
| Maximale Anzahl von Benutzergruppen | 100 |

Referenz

Die Themen in diesem Abschnitt beschreiben die Arbeit mit der Amazon-ElastiCache-API und dem ElastiCache-Abschnitt der AWS CLI. In diesem Abschnitt sind auch allgemeine Fehlermeldungen und Servicemeldungen enthalten.

- [Verwenden der ElastiCache-API](#)
- [ElastiCache-API-Referenz](#)
- [ElastiCache-Abschnitt der AWS CLI-Referenz](#)
- [Amazon-ElastiCache-Fehlermeldungen](#)
- [Benachrichtigungen](#)

Verwenden der ElastiCache-API

Dieser Abschnitt enthält aufgabenorientierte Beschreibungen der Verwendung und Implementierung von ElastiCache-Operationen. Eine vollständige Beschreibung dieser Operationen finden Sie unter [Amazon-ElastiCache-API-Referenz](#)

Themen

- [Verwenden der Abfrage-API](#)
- [Verfügbare Bibliotheken](#)
- [Fehlerbehebung bei Anwendungen](#)

Verwenden der Abfrage-API

Abfrageparameter

HTTP-Query-basierte Anfragen sind HTTP-Anfragen, die das HTTP-Verb GET oder POST und einen Query-Parameter namens `action` verwenden.

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 Ganzzahlen ab 1.

Authentifizierung von Abfrageanforderungen

Sie können Abfrageanforderungen nur über HTTPS senden und müssen in jede Abfrageanforderung 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. Dieser Vorgang setzt voraus, dass Sie bei AWS registriert sind und eine Zugriffsschlüssel-ID sowie einen geheimen Zugriffsschlüssel besitzen.

Abfrageauthentifizierungsprozess

1. Der Sender erstellt eine Anforderung an AWS.
2. Der Sender berechnet die Anforderungssignatur, ein Keyed-Hashing for Hash-based Message Authentication Code (HMAC) mit einer SHA-1-Hash-Funktion, wie im nächsten Abschnitt dieses Themas beschrieben.
3. Der Sender der Anforderung sendet die Anforderungsdaten, die Signatur und die Zugriffsschlüssel-ID (den Schlüsselbezeichner des verwendeten geheimen Zugriffsschlüssels) an AWS.
4. AWS verwendet die Zugriffsschlüssel-ID, um den geheimen Zugriffsschlüssel zu ermitteln.
5. AWS erstellt anhand der Anforderungsdaten und des geheimen Zugriffsschlüssels eine Signatur und verwendet dabei denselben Algorithmus, den Sie zum Berechnen der Signatur in der Anforderung verwendet haben.
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. Sortieren Sie die UTF-8-Abfragezeichenkomponenten nach Parameternamen in natürlicher Bytereihenfolge. Die Parameter können von einer GET-URI oder von einem POST-Test stammen (wenn Content-Type = application/x-www-form-urlencoded).
 - b. Führen Sie eine URL-Codierung des Parameternamens und -werts nach folgenden Regeln durch:
 - i. Führen Sie keine URL-Codierung der nicht reservierten, von RFC 3986 definierten Zeichen durch. 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. Versehen Sie alle erweiterten UTF-8-Zeichen mit Prozentcode im Format %XY%ZA....
 - iv. Versehen Sie das Leerzeichen mit dem Prozentcode %20 (und nicht + wie in herkömmlichen Codierungsschemata).
 - c. Trennen Sie die codierten Parameternamen mit dem Gleichzeichen (=) (ASCII-Zeichen 61) von den zugehörigen codierten Werten, auch wenn der Parameterwert leer ist.
 - d. Trennen Sie diese Namen-Wert-Paare durch ein kaufmännisches Und (&) (ASCII-Code 38).
2. Erstellen Sie die Zeichenfolge für die Signatur anhand der folgenden Pseudogrammatik (das "\n" steht in ASCII für eine neue Zeile).

```
StringToSign = HTTPVerb + "\n" +  
ValueOfHostHeaderInLowercase + "\n" +  
HTTPRequestURI + "\n" +  
CanonicalizedQueryString <from the preceding step>
```

Die HTTPRequestURI-Komponente ist die absolute HTTP-Pfadkomponente des URI bis zur, jedoch nicht einschließlich der Zeichenfolge. Falls HTTPRequestURI leer ist, verwenden Sie den Schrägstrich (/).

3. Berechnen Sie eine RFC 2104-kompatible HMAC-Signatur, wobei Sie die soeben erstellte Zeichenfolge, den geheimen Zugriffsschlüssel als Schlüssel und SHA256 oder SHA1 als Hashalgorithmus verwenden.

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 vorangehende Abfragezeichenfolge berechnen Sie die HMAC-Signatur über die folgende Zeichenfolge.

```
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=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

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 Anwendungen bevorzugt unter Verwendung sprachspezifischer APIs statt der Abfrage-API entwickeln.

Diese SDKs stellen grundlegende Funktionen bereit, die nicht in den APIs enthalten sind, wie Anforderungsauthentifizierung, Anforderungswiederholungen und Fehlerbehandlung, damit die ersten Schritte einfacher sind. 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 stellt spezifische und beschreibende Fehlermeldungen bereit, um Sie bei der Behebung von Problemen während der Interaktion mit der ElastiCache-API zu unterstützen.

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. Die einfachste Möglichkeit, herauszufinden, ob ein Fehler aufgetreten ist, besteht darin, nach einem `ERROR`-Knoten in der Antwort aus der ElastiCache-API zu suchen.

Die XPath-Syntax bietet eine einfache Möglichkeit, nach einem `ERROR`-Knoten zu suchen. Darüber hinaus vereinfacht sie den Abruf von Fehlercode und Fehlermeldung. Der folgende Codeausschnitt verwendet Perl und das `XML::XPath`-Modul, um zu ermitteln, ob während 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

Die folgenden Prozesse werden empfohlen, um Probleme mit der ElastiCache-API zu diagnostizieren und zu beheben.

- Überprüfen Sie, ob ElastiCache ordnungsgemäß ausgeführt wird.

Öffnen Sie dazu ein Browserfenster und senden Sie eine Abfrageanforderung an den ElastiCache-Service (z. B. <https://elasticache.amazonaws.com>). Mit einer `MissingAuthenticationTokenException` oder einem `500 Internal Server Error` wird bestätigt, dass der Dienst verfügbar ist und auf Anforderungen reagiert.

- Überprüfen Sie die Struktur Ihrer Anforderung.

Jede ElastiCache-Operation verfügt über eine Referenzseite in der ElastiCache-API-Referenz. 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.

Für ElastiCache gibt es ein Diskussionsforum, in dem Sie nach Lösungen für Probleme suchen können, mit denen andere Entwickler bereits konfrontiert wurden. Weitere Informationen zur Anzeige des Forums finden Sie unter

<https://forums.aws.amazon.com/>.

Einrichten 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 zur AWS CLI für ElastiCache gehen.

Important

Das Amazon ElastiCache Command Line Interface (CLI) unterstützt keine ElastiCache-Verbesserungen nach API-Version 2014-09-30. Um neuere ElastiCache-Funktionalität über die Befehlszeile zu nutzen, verwenden Sie die [AWS Command Line Interface](#).

Themen

- [Voraussetzungen](#)
- [Aufrufen der Befehlszeilen-Tools](#)
- [Einrichten der Tools](#)
- [Bereitstellung von Anmeldeinformationen für die Tools](#)
- [Umgebungsvariablen](#)

Voraussetzungen

In diesem Dokument wird vorausgesetzt, dass Sie in einer Linux/UNIX- oder Windows-Umgebung arbeiten. Die Amazon-ElastiCache-Befehlszeilen-Tools funktionieren auch mit Mac OS X, wobei es sich um eine auf UNIX basierende Umgebung handelt, jedoch sind in diesem Handbuch keine besonderen Anweisungen für Mac OS X enthalten.

Als Konvention wird jeglichem Befehlszeilentext die generische Eingabeaufforderung **PROMPT>** vorangestellt. Der auf dem Rechner tatsächlich angezeigte Prompt sieht wahrscheinlich anders aus. Zudem wird **\$** zur Angabe eines Linux/UNIX-spezifischen Befehls und **C:\>** für einen Windows-spezifischen Befehl verwendet. 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. Eine JRE- oder JDK-Installation ist zulässig. Weitere Informationen zur Anzeige und zum Download von JREs für eine breite Palette an 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 Unterverzeichnis namens `bin` enthält, das wiederum die ausführbare `java`-Datei (unter Linux und UNIX) oder die ausführbare `java.exe`-Datei (unter Windows) enthält.

Einrichten der Java Home-Variablen

1. Einrichten der Java Home-Variablen.

- Geben Sie unter Linux und UNIX 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.

- Die Ausgabe unter Linux/UNIX entspricht weitgehend der Folgenden:

```
$ $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 Befehlszeilen-Tools stehen als ZIP-Datei auf der Website [ElastiCache-Entwickler-Tools](#) zur Verfügung. Diese Tools sind in Java geschrieben und beinhalten Shell-Skripts für Windows 2000/XP/Vista/Windows 7, Linux/UNIX und Mac OSX. Die ZIP-Datei ist unabhängig und erfordert keine Installation; laden Sie einfach die ZIP-Datei herunter und entpacken Sie sie in ein Verzeichnis auf Ihrem lokalen Rechner.

Einrichten der Tools

Die Befehlszeilen-Tools sind von einer Umgebungsvariablen (`AWS_ELASTICACHE_HOME`) abhängig, 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 heißt `ElastiCacheCli-A.B.nnnn` (A, B und n sind Versions-/Release-Nummern) und enthält Unterverzeichnisse namens `bin` und `lib`.

Einrichten der Umgebungsvariable `AWS_ELASTICACHE_HOME`

- Öffnen Sie ein Befehlszeilenfenster und geben Sie einen der folgenden Befehle ein, um die Umgebungsvariable `AWS_ELASTICACHE_HOME` einzurichten.
- Geben Sie unter Linux und UNIX 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 Ihrem System-Pfad hinzuzufügen. Der Rest dieser Anleitung geht davon aus, dass sich das BIN-Verzeichnis in Ihrem Systempfad befindet.

Das BIN-Verzeichnis der Tools Ihrem Systempfad hinzufügen

- Geben Sie die folgenden Befehle ein, um das BIN-Verzeichnis der Tools zu Ihrem System-PATH hinzuzufügen.
 - Geben Sie unter Linux und UNIX 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 Kommandozeilen-Tools benötigen den AWS-Zugriffsschlüssel und den geheimen Zugriffsschlüssel, die Ihnen mit Ihrem AWS-Konto bereitgestellt werden. Sie können sie über die Befehlszeile oder aus einer Anmeldeinformationsdatei auf Ihrem lokalen System beziehen.

Die Bereitstellung enthält eine Vorlagendatei `${AWS_ELASTICACHE_HOME}/credential-file-path.template`, die Sie an Ihre Informationen anpassen müssen. Nachfolgend finden Sie dem Inhalt der Template-Datei:

```
AWSAccessKeyId=<Write your AWS access ID>
AWSSecretKey=<Write your AWS secret key>
```

Important

Unter UNIX sollten Sie die Berechtigungen auf den Eigentümer der Anmeldeinformationsdatei beschränken:

```
$ chmod 600 <the file created above>
```

Wenn Sie die Anmeldedatei eingerichtet haben, müssen Sie die Umgebungsvariable `AWS_CREDENTIAL_FILE` setzen, damit die ElastiCache-Tools Ihre Informationen finden können.

So richten Sie die Umgebungsvariable `AWS_CREDENTIAL_FILE` ein

1. Legen Sie die -Umgebungsvariable fest:

- Aktualisieren Sie unter Linux und UNIX die Variable mit folgendem 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 Umgebungsvariablen `AWS_CREDENTIAL_FILE` unterstützen die meisten API-Tools der ElastiCache Command Line Interface die folgenden Variablen:

- `EC2_REGION` – Die zu verwendende AWS-Region.
- `AWS_ELASTICACHE_URL` – Die für den Serviceaufruf zu verwendende URL. Es ist nicht erforderlich, einen anderen regionalen Endpunkt anzugeben, wenn `EC2_REGION` angegeben ist oder der Parameter `--region` übergeben wird.

Die folgenden Beispiele zeigen, wie Sie die Umgebungsvariable `EC2_REGION` setzen, 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
```

Amazon-ElastiCache-Fehlermeldungen

Die folgenden Fehlermeldungen werden von Amazon ElastiCache zurückgegeben. Sie können auch weitere Fehlermeldungen erhalten, die von ElastiCache, anderen AWS-Services oder Redis zurückgegeben werden. Für Beschreibungen von Fehlermeldungen aus anderen Quellen als ElastiCache lesen Sie bitte die Dokumentation der Quelle, die die Fehlermeldung erzeugt.

- [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 [Amazon-ElastiCache-Limits](#) in 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 [Amazon-ElastiCache-Limits](#) in 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 für Redis](#).

Benachrichtigungen

Dieses Thema behandelt ElastiCache-Benachrichtigungen, die für Sie von Interesse sein 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 Eigenschaften oder Funktionalitäten von ElastiCache an.

Allgemeine Benachrichtigungen von ElastiCache

Derzeit gibt es keine ausstehenden ElastiCache-Benachrichtigungen, die nicht engine-spezifisch sind.

Spezifische ElastiCache-for-Redis-Benachrichtigungen

Derzeit liegen keine ausstehenden ElastiCache-for-Redis-Benachrichtigungen vor.

Dokumentationsverlauf ElastiCache für Redis

- API-Version: 2015-02-02
- Letzte Aktualisierung der Dokumentation: 27. November 2023

In der folgenden Tabelle werden wichtige Änderungen in jeder Version des ElastiCache für Redis nach März 2018 beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie den RSS-Feed abonnieren.

Kürzlich ElastiCache für Redis-Updates

| Änderung | Beschreibung | Datum |
|---|--|------------------|
| ElastiCache für Redis hat Unterstützung für zusätzliche C7gn-Knotengrößen hinzugefügt | ElastiCache for Redis hat Unterstützung für zusätzliche C7gn-Knotengrößen hinzugefügt. | 10. Januar 2024 |
| ElastiCache for Redis unterstützt jetzt die Erstellung von Serverless-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 zu <code>ElastiCacheServiceRolePolicy</code> und <code>AmazonElastiCacheFullAccess</code> hinzugefügt, um die Zuordnung von Serverless-Caches zu | 8. November 2023 |

verwalteten VPC-Endpunkten zu ermöglichen. Darüber hinaus wurden Berechtigungen hinzugefügt, um ein überarbeitetes Konsolenerlebnis mithilfe der Richtlinie `AmazonElastiCacheFullAccess` zu unterstützen.

[ElastiCache for Redis unterstützt jetzt das Ändern des Clustermodus](#)

Sie können Cluster jetzt vom deaktivierten Clustermodus (CMD) zum aktivierten Clustermodus (CME) migrieren. Weitere Informationen finden Sie unter [Ändern des Clustermodus](#).

11. Mai 2023

[ElastiCache for Redis unterstützt jetzt das Ändern der Verschlüsselungseinstellungen während der Übertragung](#)

Sie können jetzt die TLS-Konfiguration Ihrer Redis-Cluster ändern, ohne die Cluster neu erstellen oder erneut 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 for Redis unterstützt jetzt die Authentifizierung von Benutzern mit IAM](#)

Mit der IAM-Authentifizierung können Sie eine Verbindung zu ElastiCache für Redis mithilfe von AWS IAM-Identitäten authentifizieren. 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 for Redis unterstützt jetzt Redis 7](#)

Diese Version bietet mehrere neue Funktionen für Amazon ElastiCache für Redis: Redis-Funktionen, ACL-Verbesserungen und Sharded Pub/Sub. Weitere Informationen finden Sie unter [ElastiCache für Redis Version 7.0](#).

08. November 2022

[ElastiCache for Redis unterstützt jetzt IPv6](#)

ElastiCache unterstützt die Internetprotokollversionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er nur IPv4-Verbindungen, nur IPv6-Verbindungen oder sowohl IPv4- als auch IPv6-Verbindungen (Dual-Stack) akzeptiert. IPv6 wird für Workloads unterstützt, die die Redis-Engine ab Version 6.2 auf allen Instances verwenden, die auf dem [Nitro-System](#) erstellt wurden. Für den Zugriff auf ElastiCache über IPv6 fallen keine zusätzlichen Gebühren an. Weitere Informationen zu Netzwerktypen finden Sie unter [Auswählen eines Netzwerktyps](#).

07. November 2022

[ElastiCache for Redis unterstützt jetzt das native JSON-Format \(JavaScript Object Notation\).](#)

Das native JSON-Format (JavaScript Object Notation) ist eine einfache, schemalose Möglichkeit, komplexe Datensätze in Redis-Clustern zu codieren. Sie können Daten nativ im JSON-Format (JavaScript Object Notation) in Redis-Clustern speichern und darauf zugreifen und JSON-Daten aktualisieren, die in diesen Clustern gespeichert sind, ohne benutzerdefinierten Code verwalten zu müssen, um sie zu serialisieren und zu deserialisieren. Weitere Informationen finden Sie unter [Erste Schritte mit JSON.](#)

25. Mai 2022

[ElastiCache unterstützt jetzt PrivateLink](#)

AWS PrivateLink ermöglicht den privaten Zugriff auf ElastiCache API-Operationen ohne Internet-Gateway, NAT-Gerät, VPN-Verbindung oder AWS Direct-Connect-Verbindung. Weitere Informationen finden Sie unter [Amazon ElastiCache API und Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) für Redis oder [Amazon ElastiCache API und Schnittstellen-VPC-Endpunkte \(AWS PrivateLink\)](#) für Memcached.

24. Januar 2022

[ElastiCache for Redis](#)
[unterstützt jetzt Redis 6.2 und](#)
[Daten-Tiering](#)

Amazon ElastiCache for Redis 23. November 2021

führt die nächste Version der Redis-Engine ein, die von Amazon unterstützt wird ElastiCache. ElastiCache für Redis 6.2 enthält Leistungsverbesserungen für TLS-fähige Cluster, die x86-Knotentypen mit 8 vCPUs oder mehr oder Graviton2-Knotentypen mit 4 vCPUs oder mehr verwenden. ElastiCache für Redis 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 für Redis Version 6.2 \(erweitert\)](#) und [Daten-Tiering](#).

[Support für Auto Scaling](#)

ElastiCache for Redis unterstützt jetzt Auto Scaling . ElastiCache for Redis Auto Scaling ist die Möglichkeit, die gewünschten Shards oder Replikate in Ihrem ElastiCache for Redis-Service automatisch zu erhöhen oder zu verringern. ElastiCache nutzt den Application Auto Scaling-Service, um diese Funktionalität bereitzustellen. Weitere Informationen finden Sie unter [Auto Scaling ElastiCache für Redis-Cluster](#).

19. August 2021

[Support für die Bereitstellung von Redis-Slow-Protokollen](#)

ElastiCache Mit können Sie jetzt Redis SLOWLOG an eines von zwei Zielen streamen: Amazon Data Firehose oder Amazon CloudWatch Logs. Weitere Informationen finden Sie unter [Protokollauslieferung](#).

22. April 2021

[Support für die Markierung von Ressourcen und Bedingungs-schlüsseln](#)

ElastiCache unterstützt jetzt das Markieren, um Sie bei der Verwaltung Ihrer Cluster und anderer ElastiCache Ressourcen zu unterstützen. Weitere Informationen finden Sie unter [Markieren Ihrer ElastiCache Ressourcen](#). führt ElastiCache auch die Unterstützung für Bedingungs-schlüssel ein. Sie können Bedingungen angeben, die bestimmen, wie eine IAM-Richtlinie wirksam wird. Weitere Informationen finden Sie unter [Verwenden von Bedingungs-schlüssel](#).

7. April 2021

[ElastiCache ist jetzt auf AWS Outposts verfügbar](#)

[AWS Outposts](#) bieten native AWS Services, Infrastrukturen und Betriebsmodelle für praktisch jedes Rechenzentrum, jeden Co-Location-Bereich oder jede On-Premises-Einrichtung. Sie können ElastiCache auf Outposts bereitstellen, um Cache On-Premises einzurichten, zu betreiben und zu verwenden, genau wie in der Cloud. Weitere Informationen finden Sie unter [Verwendung von Outposts für Redis](#) oder [Verwenden von Outposts für Memcached](#).

8. Oktober 2020

[ElastiCache unterstützt jetzt Redis 6](#)

Amazon ElastiCache for Redis führt die nächste Version der Redis-Engine ein, die von Amazon unterstützt wird ElastiCache. Diese Version umfasst die [Authentifizierung von Benutzern mit rollenbasierter Zugriffskontrolle](#), versionslosen Support, clientseitiges Caching und erhebliche betriebliche Verbesserungen. Weitere Informationen finden Sie unter [ElastiCache für Redis Version 6.0 \(Erweitert\)](#).

7. Oktober 2020

[ElastiCache unterstützt jetzt Local Zones](#)

Eine Local Zone ist eine Erweiterung einer - AWS Region, die geografisch in der Nähe Ihrer Benutzer liegt. Sie können jede Virtual Private Cloud (VPC) von einer übergeordneten AWS Region auf Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es einer Local Zone zuweisen. Weitere Informationen finden Sie unter [Local Zones nutzen](#).

25. September 2020

[ElastiCache for Redis unterstützt jetzt die Skalierung Ihrer Redis-Cluster-Umgebung auf bis zu 500 Knoten oder 500 Shards](#)

Mit dem Redis-Cluster-Modus sind Konfigurationen möglich, mit denen Sie Ihre Daten über mehrere Shards partitionieren können, und er bietet bessere Skalierbarkeit, Leistung und Verfügbarkeit. Diese Funktion ist ab Amazon ElastiCache für Redis ab Version 5.0.6 in allen - AWS Regionen und für alle vorhandenen und neuen ElastiCache für Redis-Cluster-Umgebungen verfügbar. Weitere Informationen finden Sie unter [Redis-Knoten und -Shards](#).

13. August 2020

[ElastiCache unterstützt jetzt Berechtigungen auf Ressourcenebene](#)

Sie können jetzt den Umfang der Berechtigungen eines Benutzers einschränken, indem Sie ElastiCache Ressourcen in einer AWS Identity and Access Management (IAM)-Richtlinie angeben. Weitere Informationen finden Sie unter [Berechtigungen auf Ressourcenebene](#).

12. August 2020

[ElastiCache für Redis
fügt zusätzliche Amazon-
CloudWatch Metriken hinzu](#)

ElastiCache for Redis unterstützt jetzt neue CloudWatch Metriken, einschließlich PubSubCmds und HyperLogLogBasedCmds. Eine vollständige Liste finden Sie unter [Metriken für Redis](#).

10. Juni 2020

[ElastiCache unterstützt jetzt
die automatische Aktualisi-
erung von ElastiCache
Clustern](#)

Amazon unterstützt ElastiCache jetzt die automatische Aktualisierung von ElastiCache Clustern, nachdem das „empfohlene Anwendungsdatum“ 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 for Redis unterstützt jetzt Global Datastore for Redis](#)

Die Funktion Global Datastore für Redis bietet eine vollständig verwaltete, schnelle, zuverlässige und sichere Replikation über - AWS Regionen hinweg. Mit dieser Funktion können Sie regionsübergreifende Lesereplikat-Cluster für ElastiCache für Redis erstellen , um Lesevorgänge mit niedriger Latenz und Notfallwiederherstellung über AWS Regionen hinweg zu ermöglichen. Sie können einen globalen Datenspeicher erstellen, ändern und beschreiben. Sie können auch AWS Regionen zu Ihrem globalen Datenspeicher hinzufügen oder daraus entfernen und eine - AWS Region als primäre Region innerhalb eines globalen Datenspeichers hochstufen. Weitere Informationen finden Sie unter [AWS Regionsübergreifende Replikation mit Global Datastore](#).

16. März 2020

[ElastiCache for Redis unterstützt jetzt Redis Version 5.0.6](#)

Weitere Informationen finden Sie unter [ElastiCache für Redis Version 5.0.6 \(Erweiterung\)](#).

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. Amazon EC2-T3-Standard-Instances bieten CPU-Baseline-Leistung und die Möglichkeit, diese Leistung jederzeit zu bursten, bis das angesammelte Guthaben erschöpft ist. Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

12. November 2019

[Amazon unterstützt ElastiCache jetzt das Ändern des AUTH-Tokens auf einem vorhandenen ElastiCache für Redis-Server](#)

ElastiCache for Redis 5.0.6 ermöglicht es Ihnen jetzt, Authentifizierungstoken zu ändern, indem Sie neue Token festlegen 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 auf nicht unterstützt AWS CloudFormation. Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mit Redis AUTH](#).

30. Oktober 2019

[Amazon unterstützt ElastiCache jetzt die Online-Datenmigration von Redis auf Amazon EC2](#)

Sie können jetzt Online Migration verwenden, um Ihre Daten von selbst gehostetem Redis auf Amazon EC2 zu Amazon zu migrieren ElastiCache. Weitere Informationen finden Sie unter [Online-Migration zu ElastiCache](#).

28. Oktober 2019

[ElastiCache for Redis führt die vertikale Online-Skalierung für den Redis-Cluster-Modus ein.](#)

Sie können jetzt Ihren fragmentierten Redis-Cluster bei Bedarf hoch- oder herunterskalieren. ElastiCache für Redis ändert die Größe Ihres Clusters, indem es den Knotentyp ändert, während der Cluster weiterhin online bleibt und eingehende Anfragen bedient. Weitere Informationen finden Sie unter [Vertikales Online-Skalieren durch Ändern des Knotentyps](#).

20. August 2019

[ElastiCache for Redis ermöglicht es Benutzern jetzt, einen einzelnen Reader-Endpoint für Ihren Amazon ElastiCache -for-Redis-Cluster zu verwenden.](#)

Mit dieser Funktion können Sie den gesamten Lesedatenverkehr über einen einzigen Endpunkt auf Clusterebene an Ihren ElastiCache für Redis-Cluster weiterleiten, um den Lastausgleich und die höhere Verfügbarkeit zu nutzen. Weitere Informationen finden Sie unter [Finden von Verbindungsendpunkten](#).

13. Juni 2019

[ElastiCache for Redis ermöglicht es Benutzern jetzt, Service-Updates nach ihrem eigenen Zeitplan anzuwenden](#)

Dank dieser Funktion können Sie verfügbare Service-Updates zu einem Zeitpunkt Ihrer Wahl anwenden, nicht nur in Wartungsfenstern. Dies minimiert Serviceunterbrechungen, insbesondere bei Spitzengeschäftsabläufen, und trägt dazu bei, dass Sie konform bleiben, wenn sich Ihr Cluster in von unterstützten Compliance ElastiCache-Programmen befindet. Weitere Informationen finden Sie unter [Self-Service-Updates in 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, eine Amazon- ElastiCache Instance für eine ein- oder dreijährige Laufzeit basierend auf einem Instance-Typ und einer AWS Region zu reservieren. Weitere Informationen finden Sie unter [Verwalten der Kosten mit reservierten Knoten](#).

18. Januar 2019

[ElastiCache für Redis-Unterstützung für bis zu 250 Knoten pro Redis-Cluster](#)

Das Knoten- oder Shard-Limit kann auf maximal 250 pro ElastiCache für Redis-Cluster erhöht werden. Weitere Informationen finden Sie unter [Shards](#).

19. November 2018

[ElastiCache für Redis-Unterstützung für automatisches Failover und Backup und Wiederherstellung auf allen T2-Knoten](#)

ElastiCache for Redis bietet Unterstützung für automatisches Failover, das Erstellen von Snapshots sowie Backup und Wiederherstellung auf allen T2-Knoten. Weitere Informationen finden Sie unter [ElastiCache für Redis-Backup und -Wiederherstellung](#) und [Snapshot](#).

19. November 2018

[ElastiCache für Redis-Unterstützung für M5- und R5-Knoten](#)

ElastiCache for Redis unterstützt jetzt M5- und R5-Knoten, allgemeine und speicheroptimierte Instance-Typen, die auf dem AWS Nitro-System basieren. Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

23. Oktober 2018

[Unterstützung für eine dynamische Änderung der Anzahl von Read Replicas](#)

ElastiCache for Redis hat Unterstützung für das Hinzufügen und Entfernen von Lesereplikaten zu jedem Cluster ohne Clusterausfallzeit hinzugefügt. Weitere Informationen zu diesen und anderen Änderungen in dieser Version finden Sie unter [Ändern der Anzahl der -Replikate](#) im ElastiCache Benutzerhandbuch für Redis. Siehe auch [DecreaseReplicaCount](#) und [IncreaseReplicaCount](#) in der ElastiCache API-Referenz zu .

17. September 2018

[Zertifizierung der FedRAMP-Compliance](#)

ElastiCache für Redis ist jetzt für FedRAMP-Compliance zertifiziert. Weitere Informationen finden Sie unter [Compliance-Validierung für Amazon ElastiCache](#).

30. August 2018

[Redis-Engine-Upgrades \(Cluster-Modus aktiviert\)](#)

Amazon ElastiCache for Redis hat Unterstützung für das Upgraden von Redis-Engine-Versionen (Cluster-Modus aktiviert) hinzugefügt. Weitere Informationen finden Sie unter [Upgrade von Engine-Versionen](#).

20. August 2018

[PCI DSS Compliance-Zertifizierung](#)

ElastiCache für Redis ist jetzt für PCI-DSS-Compliance zertifiziert. Weitere Informationen finden Sie unter [Compliance-Validierung für Amazon ElastiCache](#).

5. Juli 2018

[Unterstützung für ElastiCache Redis 4.0.10](#)

ElastiCache for Redis unterstützt jetzt Redis 4.0.10, einschließlich Verschlüsselung und Online-Cluster-Größenanpassung in einer einzigen Version. Weitere Informationen finden Sie unter [ElastiCache für Redis Version 4.0.10 \(Erweitert\)](#).

14. Juni 2018

[Neu strukturiertes Benutzerhandbuch](#)

Das einzelne ElastiCache Benutzerhandbuch ist jetzt so umstrukturiert, dass es separate Benutzerhandbuch für Redis ([ElastiCache für Redis-Benutzerhandbuch](#)) und für Memcached ([ElastiCache für Memcached-Benutzerhandbuch](#)) gibt. Die Dokumentationsstruktur im Abschnitt [AWS CLI Befehlsreferenz: elasticache](#) und die [Amazon ElastiCache -API-Referenz](#) bleiben unverändert.

20. April 2018

[Unterstützung der EngineCPU Utilization-Metrik](#)

ElastiCache für Redis wurde eine neue Metrik hinzugefügt `EngineCPUUtilization`, die den Prozentsatz der CPU-Kapazität meldet, der derzeit verwendet wird. Weitere Informationen finden Sie unter [Metriken für Redis](#).

9. April 2018

In der folgenden Tabelle werden die wichtigen Änderungen am ElastiCache für Redis vor März 2018 beschrieben.

| Änderung | Beschreibung | Änderungsdatum |
|--|---|-------------------|
| Support für die Region Asien-Pazifik (Osaka – regional). | <p>ElastiCache hat Unterstützung für die Region Asien-Pazifik (Osaka – lokal) 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"> • Unterstützte Regionen • Unterstützte Cache-Knotentypen | 12. Februar 2018 |
| Support für die Europa (Paris). | <p>ElastiCache hat Unterstützung für die Region EU (Paris) hinzugefügt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> • Unterstützte Regionen • Unterstützte Cache-Knotentypen | 18. Dezember 2017 |
| Support für die Region China (Ningxia) | <p>Amazon hat Unterstützung für die Region China (Ningxia) ElastiCache hinzugefügt. Weitere Informationen finden Sie hier:</p> | 11. Dezember 2017 |

| Änderung | Beschreibung | Änderungsdatum |
|---|---|-------------------|
| | <ul style="list-style-type: none">• Unterstützte Regionen• Unterstützte Cache-Knotentypen | |
| Unterstützung für dienstverknüpfte Rollen | <p>Diese Version von hat Unterstützung für serviceverknüpfte Rollen (SLR) ElastiCache hinzugefügt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Verwenden von serviceverknüpften Rollen für Amazon ElastiCache• Richten Sie Ihre Berechtigungen ein (nur ElastiCache für neue Benutzer) | 7. Dezember 2017 |
| Unterstützung für R4-Knotentypen | <p>Diese Version von ElastiCache unterstützt jetzt R4-Knotentypen in allen AWS Regionen, die von unterstützt werden ElastiCache. 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">• Unterstützte Cache-Knotentypen• Spezifische Parameter für den Redis-Knotentyp | 20. November 2017 |

| Änderung | Beschreibung | Änderungsdatum |
|--|---|------------------|
| ElastiCache für Redis 3.2.10 und Unterstützung für Online-Resharding | <p>Amazon ElastiCache for Redis bietet Unterstützung für ElastiCache für Redis 3.2.10. ElastiCache for Redis führt auch eine Online-Cluster-Größenanpassung ein, um Shards zum Cluster hinzuzufügen oder daraus zu entfernen, während es weiterhin eingehende E/A-Anfragen bedient. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Online-Größenanpassung von Clustern• Online-Resharding und Shard-Rebalancing für Redis (Clustermodus aktiviert) | 9. November 2017 |
| HIPAA-Konformität | <p>ElastiCache für Redis Version 3.2.6 ist jetzt für die HIPAA-Zulassung zertifiziert, wenn die Verschlüsselung auf Ihrem Cluster aktiviert ist. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Konformitätsvalidierung für Amazon ElastiCache• Datensicherheit in Amazon ElastiCache | 2. November 2017 |

| Änderung | Beschreibung | Änderungsdatum |
|---|--|------------------|
| ElastiCache für Redis 3.2.6 und Unterstützung für Verschlüsselung | <p>ElastiCache fügt Unterstützung für ElastiCache Redis 3.2.6 hinzu, das zwei Verschlüsselungsfunktionen umfasst:</p> <ul style="list-style-type: none">• 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.• Bei der Verschlüsselung im Ruhezustand werden Ihre Daten auf dem Laufwerk bei Synchronisierungs- und Sicherungsoperationen verschlüsselt. <p>Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Datensicherheit in Amazon ElastiCache• Unterstützte ElastiCache-for-Redis-Versionen | 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 einer Amazon VPC behandelt.</p> <p>Weitere Informationen finden Sie unter Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC im ElastiCache -Benutzerhandbuch.</p> | 24. April 2017 |

| Änderung | Beschreibung | Änderungsdatum |
|--|--|----------------|
| 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">• Testen des automatischen Failovers im ElastiCache -Benutzerhandbuch.• TestFailover in der API-Referenz zu ElastiCache• test-failover in der AWS CLI -Referenz. | 4. April 2017 |
| Erweiterte Redis-Wiederherstellung | <p>ElastiCache fügt eine verbesserte Redis-Sicherung und -Wiederherstellung mit Größenanpassung des Clusters 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. (Bei der API und der CLI wird mit dieser Funktion statt einer unterschiedlichen Anzahl an Shards eine unterschiedliche Anzahl an Knotengruppen wiederhergestellt.) Mit diesem Update werden auch unterschiedliche Redis-Slot-Konfigurationen unterstützt. Weitere Informationen finden Sie unter Wiederherstellen aus einem Backup in einen neuen Cache.</p> | 15. März 2017 |

| Änderung | Beschreibung | Änderungsdatum |
|--|--|-------------------|
| Neue Redis-Parameter für Arbeitsspeicherverwaltung | <p>ElastiCache fügt einen neuen Redis-Parameter <code>hinzureserved-memory-percent</code> , der die Verwaltung Ihres reservierten Speichers erleichtert. Dieser Parameter ist für alle Versionen von ElastiCache für Redis verfügbar. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Verwalten von reserviertem Speicher• Neue Parameters für Redis 3.2.4 | 15. März 2017 |
| 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">• Unterstützte Regionen• Unterstützte Cache-Knotentypen | 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">• Unterstützte Regionen• Unterstützte Cache-Knotentypen | 8. Dezember 2016 |
| Unterstützung für Knotentypen M4 und R3 | <p>ElastiCache fügt Unterstützung für R3- und M4-Knotentypen in der Region Südamerika (São Paulo) und M4-Knotentypen in der Region China (Peking) hinzu. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Unterstützte Regionen• Unterstützte Cache-Knotentypen | 1. November 2016 |

| Änderung | Beschreibung | Änderungsdatum |
|---|--|------------------|
| 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">• Unterstützte Regionen• Unterstützte Cache-Knotentypen | 17. Oktober 2016 |
| Unterstützung für Redis Cluster | <p>ElastiCache fügt Unterstützung für Redis-Cluster hinzu (erweitert). Kunden, die Redis Cluster verwenden, können ihre Daten über bis zu 15 Shards (Knotengruppen) hinweg partitionieren. Jeder Shard unterstützt Replikation mit bis zu 5 Lesereplikaten pro Shard. Die automatischen Failover-Zeiten von Redis Cluster sind in etwa ein Viertel so lang wie die früherer 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">• Vergleichen von Memcached und Redis• ElastiCache für Redis-Komponenten und -Funktionen – Beachten Sie die Abschnitte zu Knoten, Shards, Clustern und Replikation.• ElastiCache-for-Redis-Terminologie | 12. Oktober 2016 |

| Änderung | Beschreibung | Änderungsdatum |
|--------------------------------------|--|----------------|
| Unterstützung für Knotentyp M4 | <p>ElastiCache fügt Unterstützung für die M4-Familie von Knotentypen in den meisten AWS Regionen hinzu, die von unterstützt werdenElastiCache. 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"> • Unterstützte Cache-Knotentypen • Spezifische Parameter für den Redis-Knotentyp | 3. August 2016 |
| 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"> • Unterstützte Cache-Knotentypen • Spezifische Parameter für den Redis-Knotentyp | 27. Juni 2016 |
| Snapshot-Export | <p>ElastiCache fügt die Möglichkeit hinzu, einen Redis-Snapshot zu exportieren, sodass Sie von außerhalb darauf zugreifen könnenElastiCache. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> • Exportieren einer Sicherung im Amazon ElastiCache -Benutzerhandbuch • CopySnapshot in der Amazon ElastiCache -API-Referenz zu | 26. Mai 2016 |
| Vertikale Skalierung von Knotentypen | <p>ElastiCache fügt die Möglichkeit hinzu, Ihren Redis-Knotentyp hochzuskalieren. Weitere Informationen finden Sie unter Skalierung ElastiCache für Redis .</p> | 24. März 2016 |
| Einfaches Upgraden der Engine | <p>ElastiCache fügt die Möglichkeit hinzu, Ihre Redis-Cache-Engine einfach zu aktualisieren. Weitere Informationen finden Sie unter Engine-Versionen und Upgrades .</p> | 22. März 2016 |

| Änderung | Beschreibung | Änderungsdatum |
|--|---|-------------------|
| Unterstützung für R3-Knotentypen | ElastiCache fügt Unterstützung für R3-Knotentypen in den Regionen China (Peking) und Südamerika (São Paulo) hinzu. Weitere Informationen finden Sie unter Unterstützte Cache-Knotentypen . | 16. März 2016 |
| Zugreifen auf ElastiCache mithilfe einer Lambda-Funktion | Es wurde ein Tutorial zum Konfigurieren einer Lambda-Funktion für den Zugriff auf ElastiCache in einer Amazon VPC hinzugefügt. Weitere Informationen finden Sie unter ElastiCache-Tutorials und -Videos . | 12. Februar 2016 |
| Unterstützung für Redis 2.8.24 | ElastiCache fügt Unterstützung für Redis Version 2.8.24 hinzu, wobei seit Redis 2.8.23 Verbesserungen hinzugefügt wurden. 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">• ElastiCache for Redis Version 2.8.24 (erweitert)• Redis 2.8 Versionshinweise | 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 Amazon-ElastiCache Konsole. | Da die neueren Redis-Versionen ein besseres und stabileres Benutzererlebnis bieten, werden die Redis-Versionen 2.6.13, 2.8.6 und 2.8.19 nicht mehr in der - ElastiCache Managementkonsole aufgeführt. Weitere Optionen und mehr Informationen finden Sie unter Unterstützte ElastiCache-for-Redis-Versionen . | 15. Dezember 2015 |

| Änderung | Beschreibung | Änderungsdatum |
|--------------------------------|---|-------------------|
| Unterstützung für Redis 2.8.23 | ElastiCache fügt Unterstützung für Redis Version 2.8.23 hinzu, wobei seit Redis 2.8.22 Verbesserungen hinzugefügt wurden. Die Verbesserungen umfassen Fehlerkorrekturen sowie Support für den neuen Parameter <code>close-on-slave-writes</code> der in aktiviertem Zustand die Verbindung zu Clients trennt, die versuchen, in ein schreibgeschütztes Replikat zu schreiben. Weitere Informationen finden Sie unter ElastiCache for Redis Version 2.8.23 (erweitert) . | 13. November 2015 |

| Änderung | Beschreibung | Änderungsdatum |
|--------------------------------|---|--------------------|
| Unterstützung für Redis 2.8.22 | <p>ElastiCache fügt Unterstützung für Redis-Version 2.8.22 mit ElastiCache zusätzlichen Verbesserungen und Erweiterungen seit Version 2.8.21 hinzu. Zu den Verbesserungen gehören:</p> <ul style="list-style-type: none">• Implementierung eines unvergabelten Speicherprozesses, der eine erfolgreiche Speicherung ermöglicht, wenn aufgrund wenig verfügbarem Arbeitsspeicher eine vergabelte Speicherung fehlschlagen würde.• Zusätzliche CloudWatch Metriken – SaveInProgress und ReplicationBytes.• Um partielle Synchronisierung zu ermöglichen, ist der Redis-Parameter <code>repl-backlog-size</code> jetzt auf alle Cluster anwendbar. <p>Eine komplette Liste der Änderungen und weitere Informationen finden Sie unter ElastiCache for Redis Version 2.8.22 (erweitert).</p> <p>Diese Dokumentation enthält eine Neuorganisation der Dokumentation und das Entfernen der ElastiCache Command Line Interface (CLI)-Dokumentation. Informationen zur Verwendung der Befehlszeile finden Sie in der AWS Befehlszeile für ElastiCache.</p> | 28. September 2015 |
| Unterstützung für Redis 2.8.21 | <p>ElastiCache fügt Unterstützung für Redis-Version 2.8.21 und Redis-Verbesserungen seit Version 2.8.19 hinzu. Diese Redis-Version umfasst verschiedene Fehlerkorrekturen. Weitere Informationen finden Sie unter Redis 2.8 Versionshinweise.</p> | 29. Juli 2015 |

| Änderung | Beschreibung | Änderungsdatum |
|--|--|----------------|
| Neues Thema: Zugriff ElastiCache von außerhalb AWS | Neues Thema zum Zugriff auf ElastiCache Ressourcen von außerhalb von hinzugefügt AWS. Weitere Informationen finden Sie unter Zugriff auf ElastiCache von außerhalb von AWS . | 9. Juli 2015 |
| Nachrichten über Knotenersatz wurden hinzugefügt | ElastiCache fügt drei Nachrichten zum geplanten Knotenaustausch hinzu: ElastiCache:NodeReplacementScheduled, ElastiCache: NodeReplacementRescheduledund ElastiCache:NodeReplacementCanceled. Weitere Informationen und Maßnahmen, die Sie ergreifen können, wenn für einen Knoten ein Austausch geplant ist, finden Sie unter ElastiCache Ereignisbenachrichtigungen und Amazon SNS . | 11. Juni 2015 |

| Änderung | Beschreibung | Änderungsdatum |
|--|---|-----------------|
| Support für Redis v. 2.8.19. | <p>ElastiCache fügt Unterstützung für Redis Version 2.8.19 und Redis-Verbesserungen seit Version 2.8.6 hinzu. Diese Unterstützung enthält Unterstützung für:</p> <ul style="list-style-type: none"> • Die HyperLogLog Datenstruktur mit den Redis-Befehlen PFADD, PFCOUNT und PFMERGE. • Lexikografische Bereichsabfragen mit den neuen Befehlen ZRANGEBYLEX, ZLEXCOUNT und ZREMRANGEBYLEX • Es wurde eine Reihe von Fehlerkorrekturen vorgenommen, insbesondere um zu verhindern, dass ein Primärknoten veraltete Daten an Replikationsknoten sendet, indem der primäre SYNC fehlschlägt, wenn ein Hintergrundspeicherprozess (bgsave) unerwartet beendet wird. <p>Weitere Informationen zu HyperLogLog finden Sie unter Redis neue Datenstruktur: HyperLogLog.</p> <p>Weitere Informationen zu PFADD, PFCOUNT und PFMERGE finden Sie in der Redis-Dokumentation und klicken Sie auf HyperLogLog.</p> | 11. März 2015 |
| Unterstützung von Kostenzuordnungs-Tags | <p>ElastiCache fügt Unterstützung für Kostenzuordnungs-Tags hinzu. Weitere Informationen finden Sie unter Überwachung von Kosten mit Kostenzuordnungs-Tags.</p> | 9. Februar 2015 |
| Unterstützung für die Region AWS GovCloud (USA-West) | <p>ElastiCache fügt Unterstützung für die Region AWS GovCloud (USA-West) (us-gov-west-1) hinzu.</p> | 29. Januar 2015 |

| Änderung | Beschreibung | Änderungsdatum |
|---|---|--------------------|
| 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 | ElastiCache fügt Unterstützung für Multi-AZ vom Primärknoten zu einem Lesereplikat in einer Redis-Replikationsgruppe hinzu. ElastiCache überwacht den Zustand der Replikationsgruppe. Wenn der primäre ausfällt, stuft ein Replikat ElastiCache automatisch zum primären Replikat hoch und ersetzt dann das Replikat. Weitere Informationen finden Sie unter Minimierung der Ausfallzeiten ElastiCache bei Redis mit Multi-AZ . | 24. Oktober 2014 |
| AWS CloudTrail Protokollierung von unterstützten API-Aufrufen | ElastiCache fügt Unterstützung für die Verwendung von hinzu AWS CloudTrail , um alle ElastiCache API-Aufrufe zu protokollieren. Weitere Informationen finden Sie unter Protokollieren von Amazon-ElastiCache-API-Aufrufen mithilfe von AWS CloudTrail . | 15. September 2014 |
| Es werden neue Instance-Größen unterstützt | ElastiCache fügt Unterstützung für zusätzliche Allzweck-Instances (T2) hinzu. Weitere Informationen finden Sie unter Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen . | 11. September 2014 |
| Es werden neue Instance-Größen unterstützt | ElastiCache fügt Unterstützung für zusätzliche Allzweck-Instances (M3) und speicheroptimierte Instances (R3) hinzu. Weitere Informationen finden Sie unter Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen . | 1. Juli 2014 |

| Änderung | Beschreibung | Änderungsdatum |
|---|---|----------------|
| Sicherung und Wiederherstellung für Redis-Cluster | <p>In dieser Version ElastiCache erlaubt es Kunden, Snapshots ihrer Redis-Cluster zu erstellen und neue Cluster mit diesen Snapshots zu erstellen . Eine Sicherung ist eine Kopie des Clusters zu einem bestimmten Zeitpunkt und umfasst Cluster-Metadaten und alle Daten im Redis-Cache. Sicherungen werden in Amazon S3 gespeichert. Kunden können die Daten von einem Snapshot jederzeit in einem neuen Cluster wiederherstellen. Weitere Informationen finden Sie unter Snapshot und Wiederherstellung.</p> | 24. April 2014 |
| Redis 2.8.6 | <p>ElastiCache unterstützt zusätzlich zu Redis 2.6.13 Redis 2.8.6. Mit Redis 2.8.6 können Kunden die Resilienz und Fehlertoleranz von Lesereplikaten verbessern, mit Support für partielle Neusynchronisierung und einer benutzerdefinierten Mindestanzahl an Lesereplikaten, die jederzeit verfügbar sein müssen. Redis 2.8.6 bietet auch vollen Support für publish-and-subscribe, bei dem Clients über Ereignisse informiert werden können, die auf dem Server auftreten.</p> | 13. März 2014 |

| Änderung | Beschreibung | Änderungsdatum |
|---|--|-------------------|
| Redis-Cache-Engine | <p>ElastiCache bietet zusätzlich zu Memcached auch Redis-Cache-Engine-Software an. Kunden, die derzeit Redis verwenden, können einen neuen ElastiCache Redis-Cache-Cluster mit ihren vorhandenen Daten aus einer Redis-Snapshot-Datei „einrichten“, wodurch die Migration zu einer verwalteten ElastiCache Umgebung erleichtert wird.</p> <p>Um Redis-Replikationsfunktionen zu unterstützen, unterstützt die ElastiCache -API jetzt Replikationsgruppen. Kunden können eine Replikationsgruppe mit einem primären Redis-Cache-Knoten erstellen und mindestens einen Lesereplikat-Knoten hinzufügen, der automatisch mit den Cache-Daten im Primärknoten synchronisiert bleibt. Leseintensive 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 |
| Support für standardmäßige Amazon Virtual Private Cloud (VPC) | <p>In dieser Version ElastiCache ist vollständig in Amazon Virtual Private Cloud (VPC) integriert. Standardmäßig werden Cache-Cluster für neue Kunden in einer Amazon VPC erstellt. Weitere Informationen finden Sie unter Amazon VPCs und ElastiCache Sicherheit.</p> | 8. Januar 2013 |

| Änderung | Beschreibung | Änderungsdatum |
|--|--|-------------------|
| Support für Amazon Virtual Private Cloud (VPC) | In dieser Version können ElastiCache Cluster in Amazon Virtual Private Cloud (VPC) gestartet werden. Standardmäßig werden die Cache-Cluster neuer Kunden automatisch in einer Amazon VPC erstellt; bestehende Kunden können in ihrem eigenen Tempo zu Amazon VPC migrieren . Weitere Informationen finden Sie unter Amazon VPCs und ElastiCache Sicherheit . | 20. Dezember 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 Amazon ElastiCache -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.