



ElastiCache (Memcached) Benutzerhandbuch

Amazon ElastiCache



API-Version 2015-02-02

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

Amazon ElastiCache: ElastiCache (Memcached) Benutzerhandbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. 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 (Memcached)?	1
Serverless-Caching	1
Selbst entworfene Cluster	2
Funktionsweise	2
Cache- und Caching-Engines	2
Anwendungsfälle	8
In-Memory-Datastore	8
ElastiCache Kundenreferenzen	10
ElastiCache (Memcached) Ressourcen	10
Tools zur Verwaltung der Implementierung	12
Mit dem AWS Management Console	12
Mit dem AWS CLI	12
Verwenden des AWS SDK	12
Verwenden der API ElastiCache	12
Weitere Informationen finden Sie auch unter	13
Auswählen aus den Bereitstellungsoptionen	13
Vergleich der selbst entworfenen Caches von Memcached und Redis OSS	14
Erste Schritte mit ElastiCache (Memcached)	20
Einrichtung	20
Melde dich an für ein AWS-Konto	20
Erstellen eines Benutzers mit Administratorzugriff	21
Erteilen programmgesteuerten Zugriffs	23
Berechtigungen einrichten	24
Einrichten EC2	25
Gewähren des Netzwerkzugriffs	26
Erstellen eines Caches	27
Erstellen eines Serverless-Caches	27
Lesen und Schreiben von Daten	28
(Optional) Bereinigen	33
Nächste Schritte	34
Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC	35
Schritt 1: Erstellen Sie einen ElastiCache Cache	35
Schritt 2: Erstellen einer Lambda-Funktion	37

Schritt 3: Testen der Lambda -Funktion	42
Tutorials und Videos	43
Videos	44
Auswahl von Regionen und Availability Zones	50
Überlegungen zu Availability Zones	51
Unterstützte Regionen und Endpunkte	53
Lokalisieren Ihrer Knoten	58
Local Zones verwenden	58
Aktivieren einer Local Zone	59
Verwenden von Outposts	60
Verwenden von Outposts mit der MemCached-Konsole	60
Outposts mit der AWS CLI verwenden	62
Entwerfen eines eigenen ElastiCache-Clusters	64
ElastiCache (Memcached) Komponenten und Funktionen	65
Knoten	65
Cluster	66
AWS Regionen und Verfügbarkeitszonen	68
Endpunkte	68
Parametergruppen	69
Sicherheit	69
Subnetzgruppen	69
Ereignisse	70
Verwalten von Clustern	70
Auswählen eines Netzwerktyps	72
Daten-Tiering	74
Auto Discovery	75
Vorbereitung eines Clusters	119
Erstellen eines Clusters	126
Anzeigen der Details eines Clusters	129
Modifizieren eines Clusters	134
Neustart eines Clusters	138
Knoten zu einem Cluster hinzufügen	140
Entfernen von Knoten aus einem Cluster	146
Abbrechen ausstehender Vorgänge zum Hinzufügen oder Löschen von Knoten	152
Löschen eines Clusters	153
Zugriff auf Ihren Cluster oder die	156

Ermitteln von Verbindungsendpunkten	163
Knoten verwalten	170
ElastiCache Knotenstatus anzeigen	170
Mit Knoten verbinden	176
Unterstützte Knotentypen	179
Ersetzen von Knoten	189
Reservierte Knoten	191
Migration von Knoten der vorherigen Generation	203
Arbeiten mit ElastiCache	205
Snapshot und Wiederherstellung	205
Beschränkungen	206
Planen automatischer Backups	207
Erstellen manueller Backups	208
Erstellen einer endgültigen Sicherung	210
Beschreiben von Sicherungen	212
Kopieren eines Backups	214
Wiederherstellen aus einem Backup	216
Löschen einer Sicherung	217
Markieren von Sicherungen	218
Engine-Versionen und Upgrades	219
Unterstützte Memcached-Versionen	220
Engine-Versionen und Upgrades	224
So führen Sie ein Upgrade von Engine-Versionen aus	226
Bewährte Methoden und Caching-Strategien	227
Bewährte Methoden für Memcached-Clients	227
Unterstützte Memcached-Befehle	235
Caching-Strategien	236
Verwalten Ihrer selbst entworfenen Cluster	241
Verwaltung der Wartung	242
Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen	244
Skalierung ElastiCache (Memcached)	286
Skalierung ElastiCache (Memcached)	286
Festlegen von Skalierungslimits zur Kostenkontrolle	286
Vorskalierung mit Serverless ElastiCache	286
Skalierungsgrenzen mithilfe der Konsole festlegen und AWS CLI	288
Skalierung ElastiCache (Memcached) selbst entworfener Cluster	289

Markieren Ihrer ElastiCache-Ressourcen	293
Überwachung von Kosten mit Tags	300
Verwaltung von Stichwörtern mithilfe der AWS CLI	302
Verwaltung von Tags mithilfe der ElastiCache API	306
Amazon ElastiCache Well-Architected Lens	308
Säule „Operational Excellence“	309
Säule der Sicherheit	318
Säule der Zuverlässigkeit	325
Säule der Leistungseffizienz	331
Säule der Kostenoptimierung	343
Fehlerbehebung	350
Verbindungsprobleme	350
Fehler beim Redis OSS-Client	351
Fehlerbehebung bei hoher Latenz in Serverless ElastiCache	351
Behebung von Drosselungsproblemen in Serverless ElastiCache	353
Verwandte Themen	354
Zusätzliche Schritte zur Fehlerbehebung	355
Sicherheitsgruppen	355
Netzwerk-ACLs	356
Routing-Tabellen	357
DNS-Auflösung	358
Identifizieren von Problemen mit serverseitiger Diagnose	358
Netzwerkverbindbarkeit	364
Netzwerkbezogene Grenzen	367
CPU-Verwendung	368
Verbindungen, die von der Serverseite beendet werden	371
Clientseitige Fehlerbehebung für Amazon EC2 Instances	373
Aufschlüsselung der Zeit, die zum Abschließen einer einzelnen Anfrage benötigt wird	374
Sicherheit	377
Datenschutz	378
Datensicherheit bei Amazon ElastiCache	379
Richtlinie für den Datenverkehr zwischen Netzwerken	388
Amazon VPCs und ElastiCache Sicherheit	388
Amazon-ElastiCache-API und Schnittstellen-VPC-Endpunkte (AWS PrivateLink)	413
Subnetze und Subnetzgruppen	416
Identitäts- und Zugriffsverwaltung	425

Zielgruppe	425
Authentifizierung mit Identitäten	426
Verwalten des Zugriffs mit Richtlinien	430
So ElastiCache arbeitet Amazon mit IAM	433
Beispiele für identitätsbasierte Richtlinien	440
Fehlerbehebung	443
Zugriffskontrolle	445
Übersicht über die Verwaltung von Zugriffsberechtigungen	447
Compliance-Validierung	480
Weitere Informationen	482
Ausfallsicherheit	482
Minimieren von Ausfällen	483
Sicherheit der Infrastruktur	485
Service-Updates	485
Verwaltung von Service-Updates	486
Protokollierung und Überwachung	492
Serverless-Metriken und -Ereignisse	492
Serverless-Metriken	492
Serverless-Ereignisse	497
Metriken und Ereignisse für selbst entworfene Cluster	502
Metriken für selbst entworfene Cluster	502
Ereignisse für selbst entworfene Cluster	503
Überwachung der Nutzung	511
SNSAmazon-Ereignisüberwachung	526
Protokollieren von Amazon-ElastiCache-API-Aufrufen mithilfe von AWS CloudTrail	544
Informationen zu Amazon ElastiCache in CloudTrail	544
Grundlegendes zu den Amazon-ElastiCache-Einträgen in der Protokolldatei	545
Kontingente	550
Referenz	551
Verwenden der ElastiCache-API	551
Verwenden der Abfrage-API	551
Verfügbare Bibliotheken	555
Fehlerbehebung bei Anwendungen	555
Einrichten der AWS CLI für ElastiCache	557
Voraussetzungen	557
Aufrufen der Befehlszeilen-Tools	559

Einrichten der Tools	559
Bereitstellung von Anmeldeinformationen für die Tools	560
Umgebungsvariablen	562
Fehlermeldungen	562
Benachrichtigungen	564
Allgemeine ElastiCache Benachrichtigungen	564
ElastiCache (Memcached) Benachrichtigungen	564
der Dokumentation	566
AWS Glossar	584
.....	dlxxxv

Was ist Amazon ElastiCache (Memcached)?

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

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

Note

Amazon ElastiCache arbeitet sowohl mit den Redis OSS- 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 OSS](#) in diesem Handbuch.

Serverless-Caching

ElastiCache (Memcached) bietet serverloses Caching, wodurch das Hinzufügen und Betreiben eines Memcached-basierten Caches für Ihre Anwendung vereinfacht wird. ElastiCache (Memcached) Serverless ermöglicht es Ihnen, in weniger als einer Minute einen hochverfügbaren Cache zu erstellen, sodass Sie keine Instanzen bereitstellen oder Knoten oder Cluster konfigurieren müssen. Entwickler können einen serverlosen Cache erstellen, indem sie den Cache-Namen über die ElastiCache Konsole, das SDK oder die CLI angeben.

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

ElastiCache (Memcached) Serverless ist mit Memcached 1.6 und höher kompatibel.

Entwerfen Sie Ihren eigenen Cluster für Memcached ElastiCache

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

Das Entwerfen Ihres eigenen Clusters ElastiCache (Memcached) bietet mehr Flexibilität und Kontrolle über Ihre Cluster. Sie können beispielsweise je nach Ihren Anforderungen wählen, ob Sie einen Cluster mit Single-AZ- oder AZ-übergreifender Verfügbarkeit betreiben möchten. 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 Memcached-Cluster angewendet werden sollen.

Wenn Sie Ihren eigenen ElastiCache (Memcached) entwerfen, können Sie wählen, ob Sie Memcached 1.4 und höher ausführen möchten.

Funktionsweise

Hier finden Sie einen Überblick über die wichtigsten Komponenten eines ElastiCache (Memcached-) Deployments.

Cache- und Caching-Engines

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

Note

Amazon ElastiCache arbeitet sowohl mit den Redis OSS- 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 OSS](#) in diesem Handbuch.

Themen

- [Wie funktioniert ElastiCache \(Memcached\)](#)
- [Preisdimensionen](#)

Wie funktioniert ElastiCache (Memcached)

ElastiCache (Memcached) Serverlos

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

Note

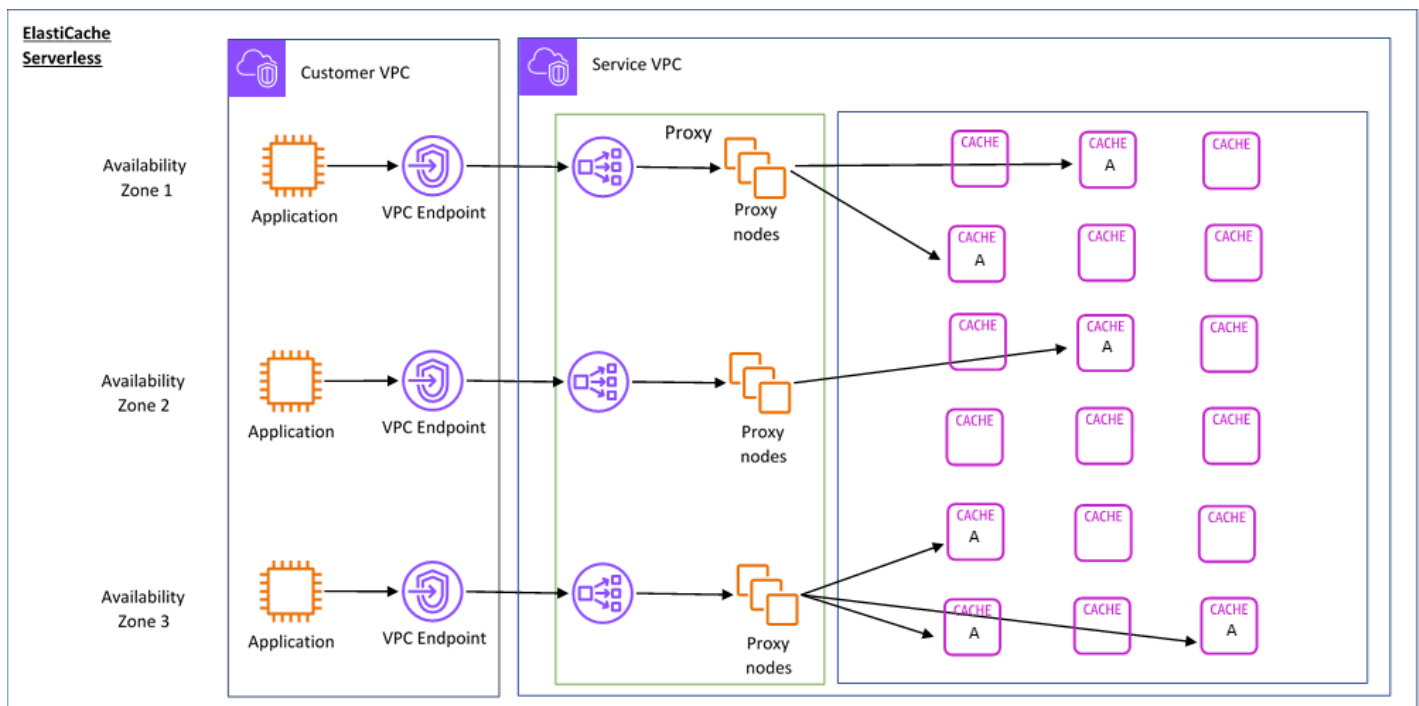
ElastiCache (Memcached) Serverless ist nur mit Memcached-Clients kompatibel, die TLS unterstützen.

Wichtigste Vorteile

- **Keine Kapazitätsplanung:** ElastiCache Serverless macht die Kapazitätsplanung überflüssig. ElastiCache Serverless überwacht kontinuierlich die Speicher-, Rechen- und Netzwerkbandbreitennutzung Ihres Caches und skaliert sowohl vertikal als auch horizontal. Dadurch kann ein Cache-Knoten an Größe zunehmen und gleichzeitig ein Aufskalierungsvorgang initiiert werden, um sicherzustellen, dass der Cache jederzeit skaliert werden kann, um Ihre Anwendungsanforderungen zu erfüllen.
- **Pay-per-use:** Bei ElastiCache Serverless zahlen Sie für die im Cache gespeicherten Daten und die Rechenleistung, die von Ihrem Workload genutzt wird. Siehe [Preisdimensionen](#).

- **Hochverfügbarkeit:** ElastiCache Serverless repliziert Ihre Daten automatisch über mehrere Availability Zones (AZ) hinweg, um eine hohe Verfügbarkeit zu gewährleisten. Es überwacht automatisch die zugrunde liegenden Cache-Knoten und ersetzt sie bei Ausfällen. Es bietet ein SLA mit einer Verfügbarkeit von 99,99 % für jeden Cache.
- **Automatische Software-Upgrades:** ElastiCache Serverless aktualisiert Ihren Cache automatisch auf die neueste Minor- und Patch-Softwareversion, ohne dass die Verfügbarkeit Ihrer Anwendung beeinträchtigt wird. Sobald eine neue Memcached-Hauptversion verfügbar ist, erhalten Sie eine Benachrichtigung.
- **Sicherheit:** Serverless verschlüsselt Daten während der Übertragung und im Ruhezustand. Sie können einen vom Service verwalteten Schlüssel oder einen eigenen kundenverwalteten Schlüssel verwenden, um Daten im Ruhezustand zu verschlüsseln.

Das folgende Diagramm zeigt, wie ElastiCache Serverless funktioniert.



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

Mit ElastiCache Serverless erhalten Sie einen einzigen DNS-Endpunkt, mit dem Ihre Anwendung eine Verbindung herstellt. Wenn Sie eine neue Verbindung zum Endpunkt anfordern, verarbeitet ElastiCache Serverless alle Cache-Verbindungen über eine Proxyschicht. Die Proxy-Schicht trägt zu

einer weniger komplexen Client-Konfiguration bei, da der Client die Cluster-Topologie nicht erneut ermitteln muss, wenn Änderungen am zugrunde liegenden Cluster vorgenommen werden. Die Proxy-Schicht besteht aus einer Reihe von Proxy-Knoten, die Verbindungen mithilfe eines Network Load Balancer verarbeiten. Wenn Ihre Anwendung eine neue Cache-Verbindung erstellt, wird die Anfrage vom Network Load Balancer an einen Proxy-Knoten gesendet. Führt Ihre Anwendung Cache-Befehle aus, dann führt der Proxy-Knoten, der mit Ihrer Anwendung verbunden ist, die Anfragen auf einem Cache-Knoten in Ihrem Cache aus. Die Proxy-Schicht abstrahiert die Cache-Cluster-Topologie und die Knoten von Ihrem Client. Dies ermöglicht ElastiCache einen intelligenten Lastenausgleich, die Skalierung und das Hinzufügen neuer Cache-Knoten, den Austausch von Cache-Knoten bei deren Ausfall und die Aktualisierung der Software auf den Cache-Knoten, ohne dass die Verfügbarkeit Ihrer Anwendung beeinträchtigt oder Verbindungen zurückgesetzt werden müssen.

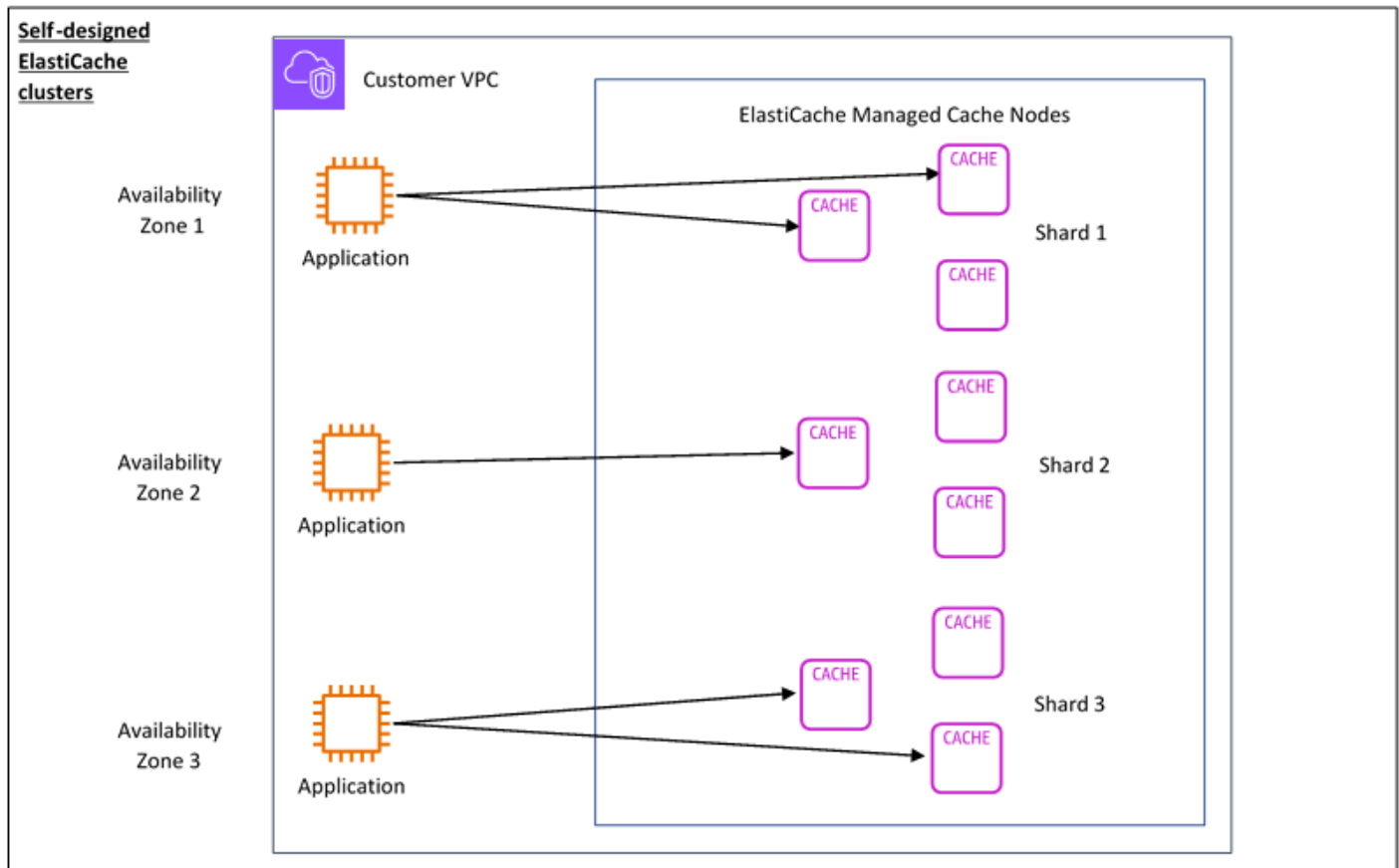
Selbst entworfene Cluster ElastiCache

Sie können Ihre eigenen ElastiCache Cluster entwerfen, indem Sie eine Cache-Knotenfamilie, Größe und Anzahl der Knoten für Ihren Cluster auswählen. Durch das Entwerfen eines eigenen Clusters haben Sie eine genauere Kontrolle über die Konfiguration und Skalierung des Clusters.

Wichtigste Vorteile

- **Entwerfen Sie Ihren eigenen Cluster:** Mit ElastiCache können Sie Ihren eigenen Cluster entwerfen und auswählen, 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 [Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen](#) 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 ihn auch horizontal skalieren, indem Sie Knoten hinzufügen.

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



Preisdimensionen

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

Datenspeicher

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

ElastiCache Verarbeitungseinheiten (eCPUs)

Sie zahlen für die Anfragen, die Ihre Anwendung auf ElastiCache Serverless in ElastiCache Processing Units (eCPUs) ausführt, einer Einheit, die sowohl vCPU-Zeit als auch übertragene Daten umfasst.

- 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 für mehrere Elemente ausgeführt werden, verbrauchen proportional mehr ECPUs. Wenn Ihre Anwendung beispielsweise einen Multiget-Befehl für 3 Elemente ausführt, verbraucht sie 3 ECPUs.
- Befehle, die für mehr Elemente ausgeführt werden und mehr Daten übertragen, verbrauchen ECPUs, die auf der höheren der beiden Dimensionen basieren. Wenn Ihre Anwendung beispielsweise den Befehl GET verwendet, 3 Elemente abrufen und 3,2 KB Daten überträgt, verbraucht sie 3,2 ECPU. Wenn sie dagegen nur 2 KB Daten überträgt, verbraucht sie 3 ECPUs.

ElastiCache Serverless gibt eine neue Metrik mit dem Namen aus, die Ihnen hilft `ElastiCacheProcessingUnits`, die von Ihrer Arbeitslast verbrauchten eCPUs zu verstehen.

Knotenstunden

Sie können einen eigenen 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.

Häufige ElastiCache Anwendungsfälle und wie ElastiCache Sie helfen können

Egal, ob es darum geht, die aktuellen Nachrichten oder einen Produktkatalog zu präsentieren oder aber Eintrittskarten für ein Event zu verkaufen: Geschwindigkeit ist das, worauf es ankommt. Der Erfolg Ihrer Website und Ihres Unternehmens hängt stark von der Geschwindigkeit ab, mit der Sie Inhalte bereitstellen.

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

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

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

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

In-Memory-Datastore

Der primäre Zweck eines Hauptspeicher-basierten Key-Value Store besteht darin, superschnellen (mit einer Latenz von unter einer Millisekunde) und kostengünstigen Zugriff auf Kopien von Daten bereitzustellen. Die meisten Datenspeicher haben Bereiche mit Daten, auf die häufig zugegriffen wird, die aber selten aktualisiert werden. Außerdem wird das Abfragen einer Datenbank immer langsamer und kostspieliger sein als das Auffinden eines Schlüssels in einem Schlüsselwertpaar-Cache. Manche Datenbankabfragen sind besonders aufwendig in der Durchführung. Ein Beispiel

dafür sind Abfragen, die Joins über mehrere Tabellen oder Abfragen mit intensiven Berechnungen beinhalten. Durch die Zwischenspeicherung solcher Abfrageergebnisse zahlen Sie den Preis für die Abfrage nur einmal. Dann können Sie die Daten schnell und mehrfach abrufen, ohne die Abfrage erneut ausführen zu müssen.

Was sollte ich zwischenspeichern?

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

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

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

Unvergänglichkeit – Per Definition sind zwischengespeicherte Daten veraltete Daten. Selbst wenn es unter bestimmten Umständen nicht veraltet ist, sollte es immer als veraltet betrachtet und behandelt werden. Um festzustellen, ob Ihre Daten für die Zwischenspeicherung geeignet sind, müssen Sie die Toleranz Ihrer Anwendung gegenüber veralteten Daten ermitteln.

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

Erwägen Sie die Zwischenspeicherung Ihrer Daten, wenn Folgendes zutrifft:

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

Weitere Informationen finden Sie hier:

- [Caching-Strategien](#) im Benutzerhandbuch ElastiCache (Memcached)

ElastiCache Kundenreferenzen

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

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

ElastiCache (Memcached) Ressourcen

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

- Leistungsmerkmale und Preise — Die [Produktdetailseite](#) bietet einen allgemeinen Produktüberblick über die wichtigsten Services und die Preisgestaltung. ElastiCache
- ElastiCache Videos — In [ElastiCache Videos](#) diesem Abschnitt finden Sie Videos, die Ihnen Amazon ElastiCache for Memcached vorstellen, allgemeine Anwendungsfälle behandeln und demonstrieren, wie Sie ElastiCache (Memcached) verwenden können, um die Latenz zu reduzieren und den Durchsatz Ihrer Anwendungen zu verbessern.
- Erste Schritte – Der [Erste Schritte mit Amazon ElastiCache \(Memcached\)](#)-Abschnitt enthält ein Beispiel, das den Prozess der Erstellung eines Cache-Clusters, der Autorisierung des Zugriffs auf den Cache-Cluster, der Herstellung einer Verbindung zu einem Cache-Knoten und dem Löschen des Cache-Clusters erläutert.

- Skalierbare Leistung — Das ElastiCache Whitepaper [Performance at Scale with Amazon](#) befasst sich mit Caching-Strategien, die es Ihrer Anwendung ermöglichen, bei jeder Größenordnung eine gute Leistung zu erzielen.

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

- [AWS Command Line Interface Dokumentation](#)

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

- [AWS CLI Dokumentation für ElastiCache](#)

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

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

- [Tools für Amazon Web Services](#)

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

- [Verwenden der ElastiCache-API](#)

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

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

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

Tools zur Verwaltung der Implementierung

Sobald Sie Ihrer Amazon EC2 EC2-Instance Zugriff auf Ihren ElastiCache Cluster gewährt haben, haben Sie vier Möglichkeiten, Ihren ElastiCache Cluster zu verwalten: das AWS Management Console, das AWS CLI For ElastiCache, das AWS SDK für ElastiCache und die ElastiCache API.

Mit dem AWS Management Console

Das AWS Management Console ist der einfachste Weg, Amazon ElastiCache (Memcached) zu verwalten. In der Konsole können Sie Cache-Cluster erstellen, Cache-Knoten hinzufügen und entfernen und andere Verwaltungsaufgaben ausführen, ohne dafür einen Code schreiben zu müssen. Die Konsole bietet auch Leistungsdiagramme von Cache-Knoten CloudWatch, die die Aktivität der Cache-Engine, die Speicher- und CPU-Auslastung sowie andere Messwerte anzeigen. Weitere Informationen finden Sie in spezifischen Themen in diesem Benutzerhandbuch.

Mit dem AWS CLI

Sie können auch das AWS Command Line Interface (AWS CLI) für verwenden ElastiCache. Das AWS CLI erleichtert das Ausführen von one-at-a-time Vorgängen, wie z. B. das Starten oder Stoppen Ihres Cache-Clusters. Sie können ElastiCache Befehle auch in einer Skriptsprache Ihrer Wahl aufrufen AWS CLI , sodass Sie sich wiederholende Aufgaben automatisieren können. Weitere Informationen zu finden Sie im AWS CLIBenutzerhandbuch und in der [AWS CLI Befehlsreferenz](#).

Verwenden des AWS SDK

Wenn Sie ElastiCache von einer Anwendung aus darauf zugreifen möchten, können Sie eines der AWS Software Development Kits (SDKs) verwenden. Die SDKs verpacken die ElastiCache API-Aufrufe und isolieren Ihre Anwendung vor den Details der API auf niedriger Ebene. ElastiCache Sie geben Ihre Anmeldeinformationen ein und die SDK-Bibliotheken übernehmen die Authentifizierung sowie das Signieren von Anforderungen. Weitere Informationen zur Verwendung der AWS SDKs finden Sie unter [Tools für Amazon Web Services](#).

Verwenden der API ElastiCache

Sie können Anwendungscode auch direkt für die ElastiCache Webservice-API schreiben. Wenn Sie die API verwenden, müssen Sie den erforderlichen Code schreiben, um Ihre HTTP-Anfragen zu erstellen und zu authentifizieren, die Ergebnisse zu analysieren und etwaige ElastiCache Fehler zu behandeln. Weitere Informationen zur API finden Sie unter [Verwenden der ElastiCache-API](#).

Weitere Informationen finden Sie auch unter

Ausführlichere Informationen zur Verwaltung Ihrer Amazon ElastiCache (Memcached) -Bereitstellung finden Sie im Folgenden:

- [Arbeiten mit ElastiCache](#)
- [Richtlinie für den Datenverkehr zwischen Netzwerken](#)
- [Protokollierung und Überwachung in Amazon ElastiCache](#)

Auswählen aus den Bereitstellungsoptionen

Amazon ElastiCache bietet zwei Bereitstellungsoptionen:

- Serverless-Caching
- Entwerfen eines eigenen Clusters

Serverless-Caching

Amazon ElastiCache Serverless vereinfacht die Cache-Erstellung und lässt sich sofort skalieren, um die anspruchsvollsten Anwendungen der Kunden zu unterstützen. Mit ElastiCache Serverless können Sie in weniger als einer Minute einen hochverfügbaren und skalierbaren Cache erstellen, sodass Sie keine Cache-Cluster-Kapazität bereitstellen, planen und verwalten müssen. ElastiCache Serverless speichert Daten automatisch redundant in drei Availability Zones und bietet ein [Service Level Agreement \(SLA\)](#) mit einer Verfügbarkeit von 99,99%. ElastiCache bietet automatische Datenreplikation zwischen AZs, sodass Replikat und benutzerdefinierte Software nicht mehr manuell verwaltet werden müssen, um sie synchron zu halten.

Selbst entworfene Cluster

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

Auswählen aus den Bereitstellungsoptionen

Wählen Sie Serverless-Caching in folgenden Fällen:

- Sie erstellen einen neuen Cache für neue oder unbekannte Workloads.
- Der Datenverkehr Ihrer Anwendung ist nicht berechenbar.
- Sie möchten den einfachsten Einstieg in die Arbeit mit einem Cache.

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

- Sie verwenden bereits ElastiCache Serverless und möchten eine genauere Kontrolle über den Knotentyp, auf dem Memcached ausgeführt wird, die Anzahl der Knoten und die Platzierung der Knoten.
- Sie erwarten nicht, dass Ihr Anwendungsdatenverkehr stark schwankt, oder Sie können die Spitzen und Einbrüche Ihres Anwendungsdatenverkehrs genau vorhersagen.
- Sie können Ihre Kapazitätsanforderungen vorhersagen, um Kosten zu kontrollieren.

Verwandte Themen:

- [Entwerfen und Verwalten eines eigenen ElastiCache-Clusters für die Memcached-Implementierung](#)

Vergleich der selbst entworfenen Caches von Memcached und Redis OSS

Amazon ElastiCache unterstützt die Memcached- und Redis OSS-Cache-Engines. 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 OSS mit einer Version von ElastiCache (Redis OSS), wenn Folgendes auf Sie zutrifft:

- ElastiCache (Redis OSS) Version 7.0 (erweitert)

[Sie möchten Redis OSS Functions, Sharded Pub/Sub oder Redis OSS ACL-Verbesserungen verwenden.](#) Weitere Informationen finden Sie unter [Redis OSS](#) Version 7.0 (erweitert).

- ElastiCache (Redis OSS) 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 (Redis OSS) Version 6.0 (erweitert)

Sie möchten Benutzer mit rollenbasierter Zugriffskontrolle authentifizieren.

Weitere Informationen finden Sie unter [Redis OSS Version 6.0 \(erweitert\)](#).

- ElastiCache (Redis OSS) Version 5.0.0 (erweitert)

Sie möchten [Redis OSS-Streams](#) verwenden, eine Protokolldatenstruktur, die es Produzenten ermöglicht, neue Elemente in Echtzeit anzuhängen, und es Verbrauchern ermöglicht, Nachrichten entweder blockierend oder nicht blockierend zu konsumieren.

Weitere Informationen finden Sie unter [Redis OSS Version 5.0.0 \(erweitert\)](#).

- ElastiCache (Redis OSS) Version 4.0.10 (erweitert)

Unterstützt sowohl die Verschlüsselung als auch das dynamische Hinzufügen oder Entfernen von Shards zu Ihrem Redis OSS-Cluster (Clustermodus aktiviert).

Weitere Informationen finden Sie unter [Redis OSS Version 4.0.10](#) (erweitert).

Die folgenden Versionen sind veraltet, haben das Ende ihrer Lebensdauer erreicht oder stehen kurz vor dem Ende.

- ElastiCache (Redis OSS) Version 3.2.10 (erweitert)

Unterstützt die Fähigkeit, Shards dynamisch zu Ihrem Redis OSS-Cluster (Clustermodus aktiviert) hinzuzufügen oder daraus zu entfernen.

 **Important**

Derzeit unterstützt ElastiCache (Redis OSS) 3.2.10 keine Verschlüsselung.

Weitere Informationen finden Sie hier:

- [Redis OSS Version 3.2.10](#) (erweitert)
- Bewährte Methoden für das Online-Resharding für Redis OSS. Weitere Informationen finden Sie im Folgenden:
 - [Bewährte Methoden: Online-Resharding](#)
 - [Online-Resharding und Shard-Rebalancing für Redis OSS \(Clustermodus aktiviert\)](#)
- [Weitere Informationen zur Skalierung von Redis OSS-Clustern finden Sie unter Skalierung.](#)
- ElastiCache (Redis OSS) Version 3.2.6 (erweitert)

Wenn Sie die Funktionalität früherer Redis OSS-Versionen sowie die folgenden Funktionen benötigen, wählen Sie ElastiCache (Redis OSS) 3.2.6:

- Verschlüsselung während der Übertragung. Weitere Informationen finden Sie unter [Amazon ElastiCache \(Redis OSS\) In-Transit Encryption](#).
- Verschlüsselung im Ruhezustand. Weitere Informationen finden Sie unter [Amazon ElastiCache \(Redis OSS\) At-Rest-Verschlüsselung](#).
- ElastiCache (Redis OSS) (Clustermodus aktiviert) Version 3.2.4

Wenn Sie die Funktionalität von Redis OSS 2.8.x sowie die folgenden Funktionen benötigen, wählen Sie Redis OSS 3.2.4 (Clustermodus):

- Sie müssen Ihre Daten auf zwei bis 500 Knotengruppen verteilen (nur im Cluster-Modus).
- Sie benötigen räumliche Indizierung (Cluster-Modus oder Nicht-Cluster-Modus).
- Sie müssen nicht mehrere Datenbanken unterstützen.
- ElastiCache (Redis OSS) (nicht geclusterter Modus) 2.8.x und 3.2.4 (erweitert)

Wenn Folgendes auf Sie zutrifft, wählen Sie Redis OSS 2.8.x oder Redis OSS 3.2.4 (nicht geclusterter Modus):

- 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 OSS (Cluster-Modus deaktiviert) und Redis OSS (Cluster-Modus aktiviert)

	Memcached	Redis OSS (Cluster-Modus deaktiviert)	Redis OSS (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 OSS (Clustermodus deaktiviert)	Redis OSS (Clustermodus 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 [Unterstützte ElastiCache \(Memcached\) Versionen](#) oder [Unterstützte ElastiCache \(Redis OSS\) Versionen](#).

Erste Schritte mit Amazon ElastiCache (Memcached)

Die Themen in diesem Abschnitt führen Sie Schritt für Schritt durch den Prozess, einen serverlosen Memcached-Cache mithilfe der Konsole zu erstellen, Zugriff darauf zu gewähren, eine Verbindung zu ihm herzustellen und schließlich zu löschen. ElastiCache

Themen

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

Einrichtung

Zum Einrichten ElastiCache:

Themen

- [Melde dich an für ein AWS-Konto](#)
- [Erstellen eines Benutzers mit Administratorzugriff](#)
- [Erteilen programmgesteuerten Zugriffs](#)
- [Richten Sie Ihre Berechtigungen ein \(nur ElastiCache für neue Benutzer\)](#)
- [Einrichten EC2](#)
- [Gewähren Sie Netzwerkzugriff von einer VPC Amazon-Sicherheitsgruppe auf Ihren Cache](#)

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/die-Anmeldung>.
2. Folgen Sie den Online-Anweisungen.

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

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

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. 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 eines Benutzers mit Administratorzugriff

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

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

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

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

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

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

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

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

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

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

Anmelden als Administratorbenutzer

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

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

Weiteren Benutzern Zugriff zuweisen

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

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

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

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

Erteilen programmgesteuerten Zugriffs

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

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

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

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	<p>(Nicht empfohlen)</p> <p>Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder AWS APIs zu signieren.</p>	<p>Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.</p> <ul style="list-style-type: none"> • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM Benutzernmelddaten im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch. • Weitere Informationen finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM Benutzer im IAM Benutzerhandbuch. AWS APIs

Verwandte Themen:

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

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

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

- Benutzer und Gruppen in AWS IAM Identity Center:

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

- Benutzer, IAM die über einen Identitätsanbieter verwaltet werden:

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

- IAMBenutzer:

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

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

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

Weitere Informationen finden Sie hier:

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

Einrichten EC2

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

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

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

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

1. Melden Sie sich bei der an AWS Command Line Interface und öffnen Sie die [EC2Amazon-Konsole](#).
2. Wählen Sie in der Navigationsleiste unter Network & Security die Option Security Groups aus.
3. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe für Ihr Amazon ausVPC. Sofern Sie keine Sicherheitsgruppe zur ElastiCache Verwendung erstellt haben, wird diese Sicherheitsgruppe als Standard bezeichnet.
4. Klicken Sie auf die Registerkarte „Eingehend“ und führen Sie anschließend folgende Schritte aus.
 - a. Wählen Sie Bearbeiten aus.
 - b. Wählen Sie Regel hinzufügen aus.
 - c. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP Regel aus.
 - d. Geben Sie in das Feld Portbereich den Wert 11211 ein.
 - e. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrem Amazon starten, eine Verbindung zu Ihrem VPC Cache herstellen kann.
 - f. Wenn Sie ElastiCache Serverless verwenden, fügen Sie eine weitere Regel hinzu, indem Sie Regel hinzufügen wählen.
 - g. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP Regel aus.

- h. Geben Sie in das Feld Portbereich den Wert 11212 ein.
- i. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrem Amazon starten, eine Verbindung zu Ihrem VPC Cache herstellen kann.
- j. Wählen Sie Speichern.

Schritt 1: Erstellen eines Caches

Der Cache, den Sie gleich starten, ist live. Er wird nicht in einer Sandbox ausgeführt. Es fallen die standardmäßigen ElastiCache-Nutzungsgebühren für den Cache an, bis Sie ihn löschen. Die Gesamtkosten werden minimal sein (üblicherweise weniger als ein Dollar), wenn Sie diese Übung in einer Sitzung durcharbeiten und den Cache löschen, sobald Sie die Übung abgeschlossen haben. Weitere Informationen zu ElastiCache-Nutzungsgebühren finden Sie unter [Amazon ElastiCache](#).

Erstellen eines Serverless-Caches

AWS Management Console

So erstellen Sie einen neuen Cache mit 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. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Memcached-Caches.
3. Wählen Sie auf der rechten Seite der Konsole Memcached-Cache erstellen aus.
4. Geben Sie in den Cache-Einstellungen einen Namen ein. Sie können optional eine Beschreibung für den Cache eingeben.
5. Lassen Sie die Standardeinstellungen ausgewählt.
6. Klicken Sie auf Erstellen, um den Cache zu erstellen.
7. Sobald sich der Cache im Status „AKTIV“ befindet, können Sie mit dem Schreiben und Lesen von Daten in den Cache beginnen.

So erstellen Sie einen neuen Cache mithilfe der 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 memcached
```

Windows

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

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

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](#).

ElastiCache Erstellt standardmäßig einen Cache in Ihrer Standard-VPC. Stellen Sie sicher, dass Ihre EC2-Instance ebenfalls in der Standard-VPC erstellt wird, damit sie eine Verbindung mit dem Cache herstellen kann.

Konfiguration

Bevor Sie beginnen, stellen Sie sicher, dass Sie die richtigen Ports für den Zugriff zur Verfügung haben.

Primärer Port: 11211

Leseoptimierter Port: 11212

Serverlose Memcached-Caches kündigen zwei Ports mit demselben Hostnamen an. Der primäre Port ermöglicht Schreib- und Lesevorgänge mit denselben Konsistenzgarantien wie OSS Memcached. Der leseoptimierte Port ermöglicht Schreibvorgänge und zusätzlich eine geringere Latenz, sodass letztendlich konsistente Lesevorgänge möglich sind.

Finden Ihres Cache-Endpunkts

AWS Management Console

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

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

AWS CLI

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

Linux

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

Windows

```
aws elasticache describe-serverless-caches ^
```

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

Herstellen einer Verbindung mithilfe von OpenSSL

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

Herstellen einer Verbindung mithilfe des Memcached-Java-Clients

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

Herstellen einer Verbindung mithilfe des Memcached-PHP-Clients

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

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

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

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

Siehe https://pymemcache.readthedocs.io/en/latest/getting_started.html

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

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

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

Siehe <https://github.com/electrode-io/memcache> und <https://www.npmjs.com/package/memcache-client>

Installieren über npm `i memcache-client`

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

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

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

Weitere Informationen finden Sie unter <https://crates.io/crates/memcache> and <https://github.com/aisk/rust-memcache>.

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

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

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

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

```
c := New(net.JoinHostPort("{cluster_endpoint}", strconv.Itoa(port)))
c.DialContext = func(ctx context.Context, network, addr string) (net.Conn, error) {
```

```
var td tls.Dialer
td.Config = &tls.Config{}
return td.DialContext(ctx, network, addr)
}
foo := &Item{Key: "foo", Value: []byte("fooval"), Flags: 123}
err := c.Set(foo)
```

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

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

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

Connect mit dem Memcached.NET-Client her () EnyimMemcachedCore

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

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

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

Schritt 3: (Optional) Bereinigen

Verwendung von AWS Management Console

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

So löschen Sie einen Cache

1. Melden Sie sich 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, die der zu löschende Cache ausführt. Es wird eine Liste mit allen Caches, in denen diese Engine ausgeführt wird, angezeigt.
3. Wählen Sie den Namen des Caches aus der Liste der Caches aus, um den zu löschenden Cache anzugeben.

Important

Sie können immer nur einen Cache auf einmal von der ElastiCache-Konsole aus löschen. Werden mehrere Caches ausgewählt, wird der Löschvorgang deaktiviert.

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

Sobald Ihr Cache in den Status DELETING wechselt, fallen für ihn keine Gebühren mehr an.

Verwendung von AWS CLI

Der folgende Code löscht den Cache my-cache.

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

Die CLI-Aktion delete-serverless-cache löscht nur einen Serverless-Cache. Wenn Sie mehrere Caches löschen möchten, rufen Sie delete-serverless-cache für jeden Serverless-Cache auf, den Sie

löschen möchten. Sie brauchen nicht zu warten, bis ein Serverless-Cache fertig gelöscht ist, bevor Sie einen anderen löschen.

Für Linux, macOS oder Unix:

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

Für Windows:

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

Weitere Informationen finden Sie im AWS CLI-Thema für ElastiCache `delete-serverless-cache`.

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

Nächste Schritte

Weitere Informationen zu ElastiCache finden Sie in folgenden Themen:

- [Arbeiten mit ElastiCache](#)
- [Skalierung ElastiCache \(Memcached\)](#)
- [Kontingente für ElastiCache](#)
- [ElastiCache Best Practices und Caching-Strategien](#)
- [Anzeigen von ElastiCache-Ereignissen](#)

Tutorial: Konfiguration einer Lambda-Funktion für den Zugriff auf Amazon ElastiCache in einer Amazon VPC

In diesem Tutorial führen Sie folgende Aufgaben aus:

- Erstellen Sie einen ElastiCache Amazon-Cache in Ihrer standardmäßigen Amazon Virtual Private Cloud (Amazon VPC) in der Region us-east-1.
- Erstellen Sie eine Lambda-Funktion für den Zugriff auf den ElastiCache Cache. Wenn Sie die Lambda-Funktion erstellen, geben Sie Subnetz-IDs in Ihrer Amazon VPC und in einer VPC-Sicherheitsgruppe an, um der Lambda-Funktion den Zugriff auf Ressourcen in Ihrer VPC zu ermöglichen. Zur Veranschaulichung generiert die Lambda-Funktion in diesem Tutorial eine UUID, schreibt diese in den Cache und ruft sie aus dem Cache ab.
- Rufen Sie die Lambda-Funktion manuell auf und überprüfen Sie, ob sie auf den ElastiCache Cache in Ihrer VPC zugegriffen hat.

Important

Das Tutorial verwendet die standardmäßige Amazon VPC in der Region „us-east-1“ in Ihrem Konto. Weitere Informationen zur Amazon VPC finden Sie unter [Erste Schritte mit der Amazon VPC](#) und im Amazon-VPC-Benutzerhandbuch.

Themen

- [Schritt 1: Erstellen Sie einen ElastiCache Cache](#)
- [Schritt 2: Erstellen einer Lambda-Funktion](#)
- [Schritt 3: Testen der Lambda -Funktion](#)

Erste Schritte

[Schritt 1: Erstellen Sie einen ElastiCache Cache](#)

Schritt 1: Erstellen Sie einen ElastiCache Cache

In diesem Schritt erstellen Sie mithilfe der CLI einen ElastiCache Amazon-Cache in der standardmäßigen Amazon Virtual Private Cloud in der Region us-east-1 in Ihrem Konto. AWS

Informationen zum Erstellen eines ElastiCache serverlosen Caches mithilfe der ElastiCache Konsole oder API finden Sie [Erstellen eines Clusters](#) im ElastiCache (Memcached) User Guide.

AWS Management Console

Führen Sie den folgenden AWS CLI Befehl aus, um einen neuen serverlosen Memcache-Cluster-Cache in der Standard-VPC in der Region us-east-1 zu erstellen.

Linux

```
aws elasticache create-serverless-cache \  
--serverless-cache-name serverlessCacheForLambda \  
--region us-east-1 \  
--engine memcached
```

Windows

```
aws elasticache create-serverless-cache ^  
--serverless-cache-name serverlessCacheForLambda ^  
--region us-east-1 ^  
--engine memcached
```

Beachten Sie, dass für das Feld „Status“ der Wert CREATING festgelegt ist. Es kann einige Minuten dauern, bis die Erstellung Ihres Clusters abgeschlossen ist. ElastiCache

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

Linux

```
aws elasticache describe-serverless-caches \  
--serverless-cache-name serverlessCacheforLambda \  
--region us-east-1
```

Windows

```
aws elasticache describe-serverless-caches ^  
--serverless-cache-name serverlessCacheforLambda ^  
--region us-east-1
```

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

Fahren Sie nach dem Erstellen des neuen Caches mit [Schritt 2: Erstellen einer Lambda-Funktion](#) fort.

Nächster Schritt:

[Schritt 2: Erstellen einer Lambda-Funktion](#)

Schritt 2: Erstellen einer Lambda-Funktion

In diesem Schritt führen Sie folgende Aufgaben aus:

1. Sie erstellen mithilfe des bereitgestellten Beispiel-Codes ein Bereitstellungspaket für die Lambda-Funktion.
2. Erstellen Sie eine IAM-Rolle (Ausführungsrolle). Wenn Sie das Bereitstellungspaket hochladen, müssen Sie diese Rolle angeben, damit Lambda die Rolle annehmen und dann die Funktion in Ihrem Namen ausführen kann. Die Berechtigungsrichtlinie gewährt AWS-Lambda Berechtigungen zum Einrichten von Elastic Network-Schnittstellen oder ENIs, damit die Lambda-Funktion auf Ressourcen in der VPC zugreifen kann. In dem Beispiel greift die Lambda-Funktion auf einen ElastiCache-Cluster in der VPC zu.
3. Erstellen Sie die Lambda-Funktion, indem Sie das Bereitstellungspaket hochladen.

Nächster Schritt

[Schritt 2.1: Das Bereitstellungspaket erstellen](#)

Schritt 2.1: Das Bereitstellungspaket erstellen

Derzeit wird der Beispielcode für die Lambda-Funktion nur in Python bereitgestellt.

Python

Der folgende Python-Beispielcode liest und schreibt ein Element in Ihren ElastiCache-Cluster. Kopieren Sie den Code in eine Datei und speichern Sie diese mit dem Namen `app.py`. Achten Sie darauf, den Wert `elasticache_config_endpoint` im Code durch die Endpunktadresse zu ersetzen, die Sie in Schritt 1 kopiert haben.

```
import uuid
```

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

elasticache_config_endpoint = "serverlesscacheforlambda-
ces85m.serverless.use1.cache.amazonaws.com"
target_port = 11211

context = ssl.create_default_context()

memcached_client = Client((elasticache_config_endpoint, target_port),
    tls_context=context)

def lambda_handler(event, context):

    # create a random UUID - this will be the sample element we add to the cache
    uuid_in = uuid.uuid4().hex

    # put the UUID to the cache
    memcached_client.set("uuid", uuid_in, expire=500, noreply=False)

    # get the item (UUID) from the cache
    result = memcached_client.get("uuid")
    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 Memcached.")
    else:
        raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
{decoded_result}")

    return "Fetched value from Memcached"
```

Dieser Code verwendet die Python-Bibliothek [pymemcache](#), um Elemente in Ihrem Cache abzulegen und abzurufen. Gehen Sie wie folgt vor, um ein Bereitstellungspaket mit pymemcache zu erstellen.

1. Erstellen Sie in Ihrem Projektverzeichnis, das die `app.py`-Quelldatei enthält, einen `package`-Ordner, in dem die `pymemcache`-Bibliothek installiert werden soll.

```
mkdir package
```

2. Installieren Sie `pymemcache` mithilfe von `pip`.

```
pip install --target ./package pymemcache
```

- Erstellen Sie eine ZIP-Datei, die die pymemcache-Bibliothek enthält. Führen Sie unter Linux oder macOS den folgenden Befehl aus: Verwenden Sie unter Windows Ihr bevorzugtes ZIP-Dienstprogramm, um eine ZIP-Datei mit der pymemcache-Bibliothek 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 unter Windows Ihr bevorzugtes ZIP-Dienstprogramm, um dem Stammverzeichnis Ihrer ZIP-Datei `app.py` hinzuzufügen.

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

Nächster Schritt

[Schritt 2.2: Eine IAM-Rolle erstellen \(Ausführungsrolle\)](#)

Schritt 2.2: Eine IAM-Rolle erstellen (Ausführungsrolle)

In diesem Schritt erstellen Sie eine AWS-Identity-and-Access-Management(IAM)-Rolle mit dem folgenden vordefinierten Rollentyp und der folgenden Berechtigungsrichtlinie für den Zugriff:

- AWS-Servicerolle vom Typ AWS Lambda – mit dieser Rolle werden AWS-Lambda-Berechtigungen erteilt, um die Rolle anzunehmen.
- AWSLambdaVPCLambdaAccessExecutionRole – dies ist die Berechtigungsrichtlinie für den Zugriff, die Sie der Rolle anfügen. Die Richtlinie gewährt Berechtigungen für die EC2-Aktionen, die AWS Lambda zum Verwalten von ENIs benötigt. Sie können diese von AWS verwaltete Richtlinie in der IAM-Konsole anzeigen.

Weitere Informationen über IAM-Benutzerrollen finden Sie unter [Roles \(Delegation and Federation\) \(Rollen \(Delegation und Verbund\)\)](#) im IAM-Benutzerhandbuch.

Verwenden Sie das folgende Verfahren, um die IAM-Rolle zu erstellen.

So erstellen Sie eine IAM-Rolle (Ausführungsrolle):

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie Roles (Rollen) und anschließend Create Role (Rolle erstellen) aus.
 - Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS-Service und unter Anwendungsfälle die Option Lambda aus. Dadurch erhält der AWS-Lambda-Service Berechtigungen, um die Rolle anzunehmen. Wählen Sie Weiter aus.
 - Suchen Sie unter Berechtigungen hinzufügen nach **AWSLambdaVPCAccessExecutionRole** und aktivieren Sie das Kontrollkästchen neben der Richtlinie.
 - Wählen Sie Weiter aus.
 - Verwenden Sie in Role Name (Rollenname) einen Namen für die Rolle, der in Ihrem AWS-Konto eindeutig ist (zum Beispiel lambda-vpc-execution-role).
 - Wählen Sie Rolle erstellen aus.
3. Kopieren Sie die Rolle ARN. Sie benötigen ihn im nächsten Schritt, wenn Sie Ihre Lambda-Funktion erstellen.

Nächster Schritt

[Schritt 2.3: Aktualisieren des Bereitstellungspakets \(Erstellen der Lambda-Funktion\)](#)

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

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

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

Verwenden Sie für die `role`-Option 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 dieser VPC ein. Sie finden diese Werte in der [Amazon-VPC-Konsole](#). Die Subnetze Ihrer Standard-VPC finden Sie, indem Sie Ihre VPCs und dann die Standard-VPC Ihres AWS-Kontos auswählen. Die Sicherheitsgruppe für diese VPC finden Sie, indem Sie unter Sicherheit die Option Sicherheitsgruppen auswählen. Stellen Sie sicher, dass die Region `us-east-1` ausgewählt ist.

Für Linux, macOS oder Unix:

```
aws lambda create-function \  
--function-name AccessMemcached \  
--region us-east-1 \  
--zip-file fileb://my_deployment_package.zip \  
--role arn:aws:iam::123456789012:role/lambda-vpc-execution-role \  
--handler app.lambda_handler \  
--runtime python3.11 \  
--timeout 30 \  
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-security-group-id \  

```

Für Windows:

```
aws lambda create-function ^  
--function-name AccessMemcached ^  
--region us-east-1 ^  
--zip-file fileb://path-to/my_deployment_package.zip ^  
--role arn:aws:iam::123456789012:role/lambda-vpc-execution-role ^  
--handler app.lambda_handler ^  
--runtime python3.11 ^  
--timeout 30 ^  
--vpc-config SubnetIds=comma-separated-vpc-subnet-ids,SecurityGroupIds=default-security-group-id ^  

```

Optional können Sie die ZIP-Datei in einen AWS-Bucket in einen Amazon-S3-Bucket in derselben AWS-Region hochladen den Bucket- und Objektnamen anschließend im vorangehenden Befehl angeben. Wie nachfolgend gezeigt, müssen Sie den `--zip-file`-Parameter durch den `--code`-Parameter ersetzen:

```
--code S3Bucket=bucket-name,S3Key=zip-file-object-key
```

Unter Verwendung der AWS-Lambda-Konsole können Sie auch die Lambda-Funktion erstellen. Wählen Sie beim Erstellen der Funktion eine VPC für das Lambda aus, und wählen Sie dann die Subnetze und Sicherheitsgruppen aus den angegebenen Feldern aus.

Nächster Schritt

[Schritt 3: Testen der Lambda -Funktion](#)

Schritt 3: Testen der Lambda -Funktion

In diesem Schritt rufen Sie die Lambda-Funktion manuell mithilfe des `invoke`-Befehls auf. Wenn die Lambda-Funktion ausgeführt wird, wird eine UUID generiert und in den ElastiCache-Cluster geschrieben, 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 (AccessMemCache) mit dem AWS-Lambda-Befehl `invoke` auf.

Für Linux, macOS oder Unix:

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

Für Windows:

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

```
Success: Inserted 05fcf2e4d6c942209acc89ea79b5b15e. Fetched  
05fcf2e4d6c942209acc89ea79b5b15e from Memcached.
```

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

ElastiCache Tutorials und Videos

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

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

ElastiCache Videos

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

Themen

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

Einführungsvideos

Die folgenden Videos stellen Ihnen Amazon vor ElastiCache.

Themen

- [AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache](#)
- [AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache](#)
- [AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache](#)
- [DAT204 — Entwicklung skalierbarer Anwendungen auf AWS NoSQL-Diensten \(re:Invent 2015\)](#)
- [DAT207 — Beschleunigung der Anwendungsleistung mit Amazon \(re:Invent 2013\) ElastiCache AWS](#)

[AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache](#)

[AWS re:Invent 2020: Was ist neu bei Amazon ElastiCache](#)

[AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache](#)

[AWS re:Invent 2019: Was ist neu bei Amazon ElastiCache](#)

[AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache](#)

[AWS re:Invent 2017: Was ist neu bei Amazon ElastiCache](#)

[DAT204 — Entwicklung skalierbarer Anwendungen auf AWS NoSQL-Diensten \(re:Invent 2015\)](#)

In dieser Sitzung besprechen wir die Vorteile von NoSQL-Datenbanken und geben einen Überblick über die wichtigsten NoSQL-Services, die von AWS Amazon DynamoDB und Amazon angeboten werden. ElastiCache Dann hören wir von zwei führenden Kunden, Expedia und Mapbox, über ihre

Anwendungsfälle und architektonischen Herausforderungen und darüber, wie sie diese mithilfe von AWS NoSQL-Diensten bewältigt haben, einschließlich Entwurfsmustern und Best Practices. 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 — Entwicklung skalierbarer Anwendungen auf AWS NoSQL-Diensten \(re:Invent 2015\)](#)

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

In diesem Video erfahren Sie, wie Sie Amazon verwenden können, ElastiCache um auf einfache Weise ein In-Memory-Caching-System bereitzustellen, um die Leistung Ihrer Anwendung zu beschleunigen. Wir zeigen Ihnen, wie Sie Amazon verwenden können ElastiCache , um die Latenz Ihrer Anwendungen zu verbessern und die Belastung Ihrer Datenbankserver zu reduzieren. Wir zeigen Ihnen auch, wie Sie eine Caching-Schicht aufbauen, die einfach zu verwalten und zu skalieren ist, wenn Ihre Anwendung wächst. In dieser Sitzung gehen wir auf verschiedene Szenarien und Anwendungsfälle ein, die von der Aktivierung von Caching profitieren können, und besprechen die von Amazon ElastiCache bereitgestellten Funktionen.

[DAT207 — Schnellere Anwendungsleistung mit Amazon ElastiCache \(re:Invent 2013\)](#)

Weiterführende Videos

Die folgenden Videos behandeln fortgeschrittenere ElastiCache Amazon-Themen.

Themen

- [Erfolgreiches Design mit den ElastiCache Best Practices von Amazon \(re:Invent 2020\)](#)
- [Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache \(re:Invent 2019\)](#)
- [Bewährte Methoden: Migration von Redis OSS-Clustern von Amazon EC2 zu ElastiCache \(re:Invent 2019\)](#)
- [Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 \(re:Invent 2018\)](#)
- [Zuverlässiges und skalierbares Redis OSS in der Cloud mit Amazon ElastiCache \(re:Invent 2018\)](#)
- [ElastiCache Deep Dive: Entwurfsmuster für In-Memory-Datenspeicher \(re:Invent 2018\)](#)
- [DAT305 — Amazon ElastiCache Deep Dive \(re:Invent 2017\)](#)
- [DAT306 — Amazon ElastiCache Deep Dive \(re:Invent 2016\)](#)
- [DAT407 — Amazon ElastiCache Deep Dive \(re:Invent 2015\)](#)
- [SDD402 — Amazon ElastiCache Deep Dive \(re:Invent 2014\)](#)

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

Erfolgreiches Design mit den ElastiCache Best Practices von Amazon (re:Invent 2020)

Angesichts des explosionsartigen Wachstums geschäftskritischer Echtzeitanwendungen, die auf Redis OSS basieren, sind Verfügbarkeit, Skalierbarkeit und Sicherheit zu den wichtigsten Überlegungen geworden. Lernen Sie bewährte Methoden kennen, um Amazon ElastiCache mit Online-Skalierung, Hochverfügbarkeit in Multi-AZ-Bereitstellungen und Sicherheitskonfigurationen auf Erfolgskurs zu bringen.

[Erfolgreiches Design mit den ElastiCache Best Practices von Amazon \(re:Invent 2020\)](#)

Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache (re:Invent 2019)

Angesichts der rasanten Zunahme der Cloud-Nutzung und der damit verbundenen neuen Szenarien benötigen Anwendungen Latenzzeiten im Mikrosekundenbereich und einen hohen Durchsatz, um Millionen von Anfragen pro Sekunde zu unterstützen. Für die Verwaltung von Daten für Echtzeitanwendungen haben sich die Entwickler bisher auf spezielle Hardware und Workarounds verlassen, z. B. auf plattenbasierte Datenbanken in Kombination mit Datenreduzierungstechniken. Diese Vorgehensweisen können kostspielig und nicht erweiterungsfähig sein. Erfahren Sie, wie Sie die Leistung von Echtzeitanwendungen steigern können, indem Sie das vollständig verwaltete In-Memory-Archiv Amazon ElastiCache für extreme Leistung, hohe Skalierbarkeit, Verfügbarkeit und Sicherheit nutzen.

[Optimieren Sie Ihre Echtzeit-Apps mit Amazon ElastiCache \(re:Invent 2019\)](#)

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

Die eigenständige Verwaltung von Redis OSS-Clustern 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 ganz einfach von selbst gehostetem Redis OSS auf Amazon EC2 auf vollständig verwaltetes Amazon ElastiCache verschieben, wobei der Clustermodus deaktiviert ist. In dieser Sitzung erfahren Sie mehr über das neue Online-Migrationstool, sehen sich eine Demo an und, was noch wichtiger ist, lernen Sie praktische Best Practices für eine reibungslose Migration zu Amazon ElastiCache kennen.

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

Skalierung einer Fantasy-Sport-Plattform mit Amazon ElastiCache und Amazon Aurora STP11 (re:Invent 2018)

Dream11 ist Indiens führendes Sport-Tech-Startup. Es hat eine wachsende Basis von mehr als 40 Millionen Nutzern, die mehrere Sportarten spielen, darunter Fantasy Cricket, Fußball und Basketball, und bedient derzeit eine Million gleichzeitige Nutzer, die drei Millionen Anfragen pro Minute mit einer Reaktionszeit von 50 Millisekunden stellen. In diesem Vortrag erklärt Amit Sharma, CTO von Dream11, wie das Unternehmen Amazon Aurora und Amazon für den Umgang mit Flash-Verkehr verwendet, der ElastiCache sich innerhalb eines Antwortfensters von 30 Sekunden verdreifachen kann. Sharma spricht auch über die Skalierung von Transaktionen ohne Sperren und erklärt, wie man den Flash-Verkehr bewältigt und dabei fünf Millionen täglich aktive Nutzer bedient. Vollständiger Titel: AWS re:Invent 2018: Skalierung einer Fantasy-Sportplattform mit Amazon ElastiCache und Amazon Aurora (STP11)

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

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

In dieser Sitzung werden die Funktionen und Verbesserungen unseres Redis OSS-kompatiblen Dienstes Amazon ElastiCache (Redis OSS) behandelt. Wir behandeln wichtige Funktionen wie Redis OSS 5, Skalierbarkeits- und Leistungsverbesserungen, Sicherheit und Compliance und vieles mehr. Wir besprechen auch zukünftige Funktionen und Kundenfallstudien.

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

ElastiCache Deep Dive: Entwurfsmuster für In-Memory-Datenspeicher (re:Invent 2018)

In dieser Sitzung bieten wir einen Blick hinter die Kulissen, um mehr über das Design und die Architektur von Amazon zu erfahren ElastiCache. Sehen Sie sich die gängigen Entwurfsmuster unserer Redis OSS- und Memcached-Angebote an und erfahren Sie, wie Kunden sie für die In-Memory-Datenverarbeitung verwenden, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. Wir überprüfen ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns.

[ElastiCache Deep Dive: Entwurfsmuster für In-Memory-Datenspeicher \(re:Invent 2018\)](#)

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

Schauen Sie hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich gängige Entwurfsmuster unserer Memcached- und Redis OSS-

Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen eingesetzt haben, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. In diesem Video gehen wir auf ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns ein.

Das Video zeigt Folgendes:

- ElastiCache (Redis OSS) Online-Resharding
- ElastiCache Sicherheit und Verschlüsselung
- ElastiCache (Redis OSS) Version 3.2.10

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

DAT306 — Amazon ElastiCache Deep Dive (re:Invent 2016)

Schauen Sie hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich gängige Entwurfsmuster unserer Memcached- und Redis OSS-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen eingesetzt haben, um die Latenz zu reduzieren und den Anwendungsdurchsatz zu verbessern. In dieser Sitzung besprechen wir ElastiCache bewährte Verfahren, Entwurfsmuster und Anti-Patterns.

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

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

Werfen Sie einen Blick hinter die Kulissen, um mehr über ElastiCache das Design und die Architektur von Amazon zu erfahren. Sehen Sie sich die gängigen Entwurfsmuster unserer Memcached- und Redis OSS-Angebote an und erfahren Sie, wie Kunden sie für In-Memory-Operationen eingesetzt und eine verbesserte Latenz und einen verbesserten Durchsatz für Anwendungen erzielt haben. In dieser Sitzung besprechen wir bewährte Verfahren, Entwurfsmuster und Anti-Pattern-Methoden im Zusammenhang mit Amazon ElastiCache.

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

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

In diesem Video untersuchen wir gängige Caching-Anwendungsfälle, die Memcached- und Redis OSS-Engines, Muster, anhand derer Sie ermitteln können, welche Engine für Ihre Anforderungen besser geeignet ist, konsistentes Hashing und vieles mehr, um schnelle, skalierbare Anwendungen zu entwickeln. Frank Wiebe, Principal Scientist bei Adobe, beschreibt, wie Adobe Amazon nutzt, ElastiCache um das Kundenerlebnis zu verbessern und sein Geschäft zu skalieren.

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

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

In diesem Video untersuchen wir Caching, Caching-Strategien, Skalierung und Überwachung. Wir vergleichen auch die OSS-Engines Memcached und Redis. Während dieser Sitzung besprechen wir auch Best Practices und Designmuster im Zusammenhang mit Amazon ElastiCache.

[DAT307 — Tiefer Einblick in die ElastiCache Architektur und Entwurfsmuster von Amazon \(AWS re:Invent 2013\)](#).

Auswahl von Regionen und Availability Zones

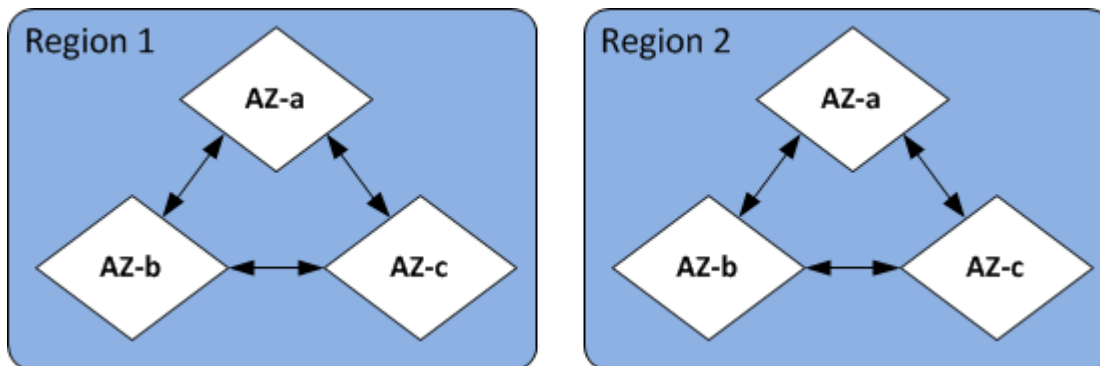
AWS Cloud-Computing-Ressourcen sind in hochverfügbaren Rechenzentrumseinrichtungen untergebracht. Die Einrichtungen dieser Rechenzentren befinden sich an verschiedenen Standorten, um die Skalierbarkeit und Zuverlässigkeit zu erhöhen. Diese Standorte sind nach Regionen und Availability Zones kategorisiert.

AWS Die Regionen sind groß und weit über verschiedene geografische Standorte verteilt. Availability Zones sind unterschiedliche Standorte innerhalb einer AWS Region, die so konzipiert sind, dass sie von Ausfällen in anderen Availability Zones isoliert sind. Sie bieten kostengünstige Netzwerkkonnektivität mit niedriger Latenz zu anderen Availability Zones in derselben AWS Region.

⚠ Important

Jede Region ist komplett eigenständig. Jede ElastiCache Aktivität, die Sie initiieren (z. B. das Erstellen von Clustern), wird nur in Ihrer aktuellen Standardregion ausgeführt.

Wenn Sie einen Cluster in einer bestimmten Region erstellen oder damit arbeiten möchten, müssen Sie den entsprechenden regionalen Service-Endpunkt wählen. Informationen zu Service-Endpunkten finden Sie unter [Unterstützte Regionen und Endpunkte](#).



Regionen und Availability Zones

Themen

- [Überlegungen zu Availability Zones](#)
- [Unterstützte Regionen und Endpunkte](#)
- [Lokalisieren Ihrer Knoten](#)

- [Verwenden von lokalen Zonen mit ElastiCache](#)
- [Verwenden von Outposts](#)

Überlegungen zu Availability Zones

Die Verteilung Ihrer Memcached-Knoten auf mehrere Availability Zones innerhalb einer Region schützt Sie vor den Auswirkungen eines katastrophalen Ausfalls, z. B. eines Stromausfalls innerhalb einer Availability Zone.

Serverless-Caching

ElastiCache Beim serverlosen Caching wird ein hochverfügbarer Cache erstellt, der sich über mehrere Availability Zones erstreckt. Sie können beim Erstellen Ihres Caches Subnetze aus verschiedenen Availability Zones und derselben VPC angeben oder ElastiCache Subnetze automatisch aus Ihrer Standard-VPC auswählen.

Entwerfen Sie Ihren eigenen ElastiCache (Memcached-) Cluster

Ein Memcached-Cluster kann bis zu 300 Knoten haben. Wenn Sie Knoten erstellen oder zu Ihrem Memcached-Cluster hinzufügen, können Sie eine einzige Availability Zone für all Ihre Knoten angeben, die Auswahl einer einzigen Availability Zone für alle Ihre Knoten zulassen ElastiCache , die Availability Zones für jeden Knoten angeben oder die Auswahl einer Availability Zone für jeden Knoten ermöglichen ElastiCache . Wenn Sie neue Knoten zu einem bestehenden Memcached-Cluster hinzufügen, können diese in verschiedenen Availability Zones erstellt werden. Sobald ein Cache-Knoten erstellt wurde, kann seine Availability Zone nicht mehr geändert werden.

Wenn Sie möchten, dass die Knoten eines Clusters in einem einzelnen Availability Zone-Cluster auf mehrere Availability Zones verteilt sind, ElastiCache können Sie neue Knoten in den verschiedenen Availability Zones erstellen. Anschließend können Sie einige oder alle ursprünglichen Cache-Knoten löschen. Wir empfehlen diesen Ansatz.

Migration von Memcached-Knoten aus einer einzelnen Availability Zone in mehrere Availability Zones

1. Ändern Sie Ihren Cluster, indem Sie neue Cache-Knoten in den Availability Zones erstellen, in denen Sie sie benötigen. Führen Sie in Ihrer Anforderung folgende Schritte aus:
 - Setzen Sie AZMode (CLI: `--az-mode`) zu `cross-az`.
 - Setzen Sie NumCacheNodes (CLI: `--num-cache-nodes`) auf die Anzahl der aktuell aktiven Cache-Knoten plus die Anzahl der neuen Cache-Knoten, die Sie erstellen möchten.

- Setzen Sie `NewAvailabilityZones` (CLI: `--new-availability-zones`) in eine Liste der Zonen, in denen Sie die neuen Cache-Knoten erstellen wollen. Um die Availability Zone für jeden neuen Knoten ElastiCache bestimmen zu lassen, geben Sie keine Liste an.
- Setzen Sie `ApplyImmediately` (CLI: `--apply-immediately`) auf `true`.

 Note

Wenn Sie keine automatische Erkennung verwenden, stellen Sie sicher, dass Sie Ihre Client-Anwendung mit den neuen Cache-Knoten-Endpunkten aktualisieren.

Bevor Sie mit dem nächsten Schritt fortfahren, müssen Sie sicherstellen, dass die Memcached-Knoten vollständig erstellt und verfügbar sind.

2. Ändern Sie den Cluster, indem Sie die Knoten entfernen, die Sie in der ursprünglichen Availability Zone nicht mehr benötigen. Führen Sie in Ihrer Anforderung folgende Schritte aus:
 - Setzen Sie `NumCacheNodes` (CLI: `--num-cache-nodes`) auf die Anzahl der aktiven Cache-Knoten, die Sie nach der Anwendung dieser Änderung möchten.
 - Setzen Sie `CacheNodeIdsToRemove` (CLI: `--nodes-to-remove`) in eine Liste der Cache-Knoten, die Sie aus dem Cluster entfernen möchten.

Die Anzahl der aufgelisteten Cache-Knoten-IDs muss der Anzahl der derzeit aktiven Knoten minus dem Wert in `NumCacheNodes` entsprechen.

- (Optional) Setzen Sie `ApplyImmediately` (CLI: `--apply-immediately`) auf `true`.

Wenn Sie `ApplyImmediately` (CLI: `--apply-immediately`) nicht auf `true` gesetzt ist, werden die Knotenlöschungen in Ihrem nächsten Wartungsfenster stattfinden.

Unterstützte Regionen und Endpunkte

Amazon ElastiCache ist in mehreren AWS Regionen verfügbar. Das bedeutet, dass Sie ElastiCache Cluster an Standorten starten können, die Ihren Anforderungen entsprechen. Sie können beispielsweise in der AWS Region starten, die Ihren Kunden am nächsten ist, oder in einer bestimmten AWS Region, um bestimmte gesetzliche Anforderungen zu erfüllen.

Jede -Region ist darauf ausgelegt, vollständig von den anderen -Regionen getrennt zu sein. Innerhalb jeder Region gibt es mehrere Availability Zones (AZ). ElastiCache Serverlose Caches replizieren Daten automatisch über mehrere Availability Zones hinweg (außer wenn Daten in zwei Availability Zones repliziert werden)us-west-1, um eine hohe Verfügbarkeit zu gewährleisten. Beim Entwerfen Ihres eigenen ElastiCache Clusters können Sie wählen, ob Sie Ihre Knoten in verschiedenen AZs starten möchten, um 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 ElastiCache dies unterstützt wird

Regionsname/Region	Endpunkt	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	

Regionsname/Region	Endpoint	Protokoll	
Region Kanada (Zentral) ca-central-1	elasticache.ca- central-1.am azonaws.com	HTTPS	
Region Kanada (West) ca-west-1	elasticache.ca- west-1.amazo naws.com	HTTPS	
Asien-Pazifik (Jakarta) ap-southeast-3	elasticache.ap- southeast-3. amazonaws.com	HTTPS	
Region Asien-Pazifik (Mumbai) ap-south-1	elasticache.ap- south-1.amaz onaws.com	HTTPS	
Region Asien-Pazifik (Hyderabad) ap-south-2	elasticache.ap- south-2.amaz onaws.com	HTTPS	
Region Asien-Pazifik (Tokio) ap-northeast-1	elasticache.ap- northeast-1. amazonaws.com	HTTPS	
Region Asien-Pazifik (Seoul) ap-northeast-2	elasticache.ap- northeast-2. amazonaws.com	HTTPS	
Region Asien-Pazifik (Osaka) ap-northeast-3	elasticache.ap- northeast-3. amazonaws.com	HTTPS	

Regionsname/Region	Endpunkt	Protokoll	
Region Asien-Pazifik (Singapur) ap-southeast-1	elasticache.ap- southeast-1. amazonaws.com	HTTPS	
Region Asien-Pazifik (Sydney) ap-southeast-2	elasticache.ap- southeast-2. amazonaws.com	HTTPS	
Region Europa (Frankfurt) eu-central-1	elasticache.eu- central-1.am azonaws.com	HTTPS	
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	

Regionsname/Region	Endpunkt	Protokoll
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
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

Regionsname/Region	Endpunkt	Protokoll
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
AWS GovCloud (US- West) us-gov-west-1	elasticache.us- gov-west-1.a mazonaws.com	HTTPS
AWS GovCloud (US- Ost) us-gov-east-1	elasticache.us- gov-east-1.a mazonaws.com	HTTPS

Informationen zur Nutzung von AWS GovCloud (USA) mit ElastiCache finden Sie unter [Dienste in der Region AWS GovCloud \(USA\): ElastiCache](#).

Einige Regionen unterstützen eine Untermenge von Knotentypen. Eine Tabelle der unterstützten Knotentypen nach AWS Regionen finden Sie unter [Unterstützte Knotentypen nach AWS -Region](#).

Eine Tabelle mit AWS Produkten und Dienstleistungen nach Regionen finden Sie unter [Produkte und Dienstleistungen nach Regionen](#).

Lokalisieren Ihrer Knoten

Amazon ElastiCache unterstützt die Lokalisierung aller Knoten eines Clusters in einer oder mehreren Availability Zones (AZs). Wenn Sie sich außerdem dafür entscheiden, Ihre Knoten in mehreren AZs zu lokalisieren (empfohlen ElastiCache), können Sie entweder die AZ für jeden Knoten auswählen oder zulassen ElastiCache, dass sie für Sie ausgewählt werden.

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.

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](#)

Verwenden von lokalen Zonen mit ElastiCache

Eine lokale Zone ist eine Erweiterung einer AWS Region, die sich geografisch in der Nähe Ihrer Benutzer befindet. Sie können jede Virtual Private Cloud (VPC) von einer übergeordneten AWS Region auf eine Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es der lokalen Zone zuweisen. Wenn Sie ein Subnetz in einer lokalen Zone erstellen, wird Ihre VPC auf diese lokale Zone erweitert. Das Subnetz in der lokalen Zone funktioniert genauso wie andere Subnetze in Ihrer VPC.

Mithilfe von Local Zones können Sie Ressourcen wie einen ElastiCache Cluster an mehreren Standorten in der Nähe Ihrer Benutzer platzieren.

Wenn Sie einen ElastiCache Cluster erstellen, können Sie ein Subnetz in einer lokalen Zone auswählen. Local Zones haben ihre eigenen Verbindungen mit dem Internet und unterstützen AWS Direct Connect. Daher können Ressourcen, die in einer lokalen Zone erstellt wurden, von lokalen Benutzern mit Kommunikationen mit sehr geringer Latenz genutzt werden. Weitere Informationen finden Sie unter [AWS -Local-Zones](#).

Eine lokale Zone wird beispielsweise `us-west-2-lax-1a` durch einen AWS Regionalcode gefolgt von einer Kennung dargestellt, die den Standort angibt.

Zu diesem Zeitpunkt sind die verfügbaren Local Zones `us-west-2-lax-1a` und `us-west-2-lax-1b`.

Die folgenden Einschränkungen gelten ElastiCache für Local Zones:

- 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 lokalen Zone.

Wenn Sie eine ElastiCache Subnetzgruppe erstellen, wählen Sie die Availability Zone-Gruppe für die Lokale Zone aus.

Weitere Informationen finden Sie im ElastiCache Benutzerhandbuch unter [Erstellen einer Subnetzgruppe](#).

4. Erstellen Sie einen ElastiCache (Memcached-) Cluster, der das ElastiCache Subnetz in der lokalen Zone verwendet.

Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

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.
- 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 MemCached-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 Memcached Caches aus.
3. Wählen Sie Memcached-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 On-premises 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 Memcached-Einstellungen:
 - a. Name: Geben Sie einen Namen für den Memcached-Cluster ein
 - b. Beschreibung: Geben Sie eine Beschreibung für den Memcached-Cluster ein.
 - c. Kompatibilität der Engine-Version: Die Engine-Version basiert auf der AWS Outpost-Region
 - d. Port: Akzeptieren Sie den Standardport 11211. 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. 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 Knoten: Geben Sie die Anzahl der Knoten ein, die Sie in Ihrem Cluster verwenden möchten.
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.

- 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 Memcached-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 \  
  \
```

Entwerfen und Verwalten eines eigenen ElastiCache-Clusters für die Memcached-Implementierung

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 \(Memcached\) Komponenten und Funktionen](#)
- [Verwalten von Clustern](#)
- [Knoten verwalten](#)

ElastiCache (Memcached) Komponenten und Funktionen

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

Themen

- [ElastiCache Knoten](#)
- [ElastiCache \(Memcached\) -Cluster](#)
- [AWS Regionen und Verfügbarkeitszonen](#)
- [ElastiCache \(Memcached\) Endpunkte](#)
- [ElastiCache Parametergruppen](#)
- [ElastiCache Sicherheit](#)
- [ElastiCache Subnetzgruppen](#)
- [ElastiCache \(Memcached\) 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. Auf jedem Knoten wird eine Instance von Memcached ausgeführt. 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 \(Memcached\)](#).

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. Sie können reservierte Knoten auch zu einem stark reduzierten Stundensatz kaufen. 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, und pay-as-you-go Knoten für die Zeiten erwerben, in denen Sie gelegentlich Knoten hinzufügen müssen. Weitere Informationen zu reservierten Knoten finden Sie unter [Reservierte Knoten für ElastiCache](#).

Die Memcached-Engine unterstützt die Funktion "Auto Discovery". Auto Discovery oder automatische Erkennung bezeichnet die Fähigkeit von Client-Programmen, alle Knoten in einem Cache-Cluster automatisch zu identifizieren und eine Verbindung zu allen Knoten herzustellen und aufrechtzuerhalten. Mit der Funktion "Auto Discovery" muss die Anwendung die Verbindung zu individuellen Knoten nicht manuell herstellen. Stattdessen stellt die Anwendung eine Verbindung zu einem Konfigurationsendpunkt her. Der DNS-Eintrag für den Konfigurationsendpunkt enthält die CNAME-Einträge für jeden der Cache-Knotenendpunkte. Durch die Verbindung mit dem Konfigurationsendpunkt verfügt die Anwendung sofort über Informationen zu allen Knoten im Cluster und kann zu allen eine Verbindung herstellen. Die einzelnen Endpunkte der Cache-Knoten müssen nicht fest in der Anwendung codiert werden. Weitere Informationen finden Sie unter [Auto Discovery](#).

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

ElastiCache (Memcached) -Cluster

Ein Memcached-Cluster ist eine logische Gruppierung einer oder mehrerer [ElastiCache Knoten](#). In einem Memcached-Cluster sind die Daten über die Knoten hinweg partitioniert.

Viele ElastiCache Operationen sind auf Cluster ausgerichtet:

- Erstellen eines Clusters
- Modifizieren eines Clusters
- 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.

Wenn Sie diese Grenzwerte überschreiten müssen, stellen Sie Ihre Anfrage über das [Antragsformular für ElastiCache Amazon-Cache-Knoten](#).

- [Minimieren von Ausfällen](#)

Informationen zur Verbesserung der Fehlertoleranz von Clustern.

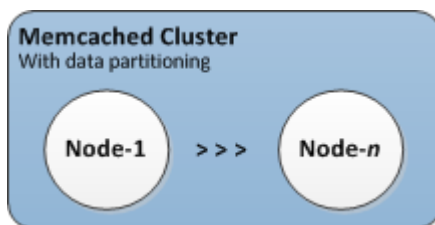
Typische Cluster-Konfigurationen

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

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

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

Wenn sich die Anforderungen an den Memcached-Cluster ändern, können Sie ihn durch Hinzufügen oder Entfernen von Knoten skalieren. Dabei werden die Daten entsprechend der neuen Anzahl an Knoten neu partitioniert. Zur Partitionierung der Daten empfehlen wir, konsistentes Hashing zu verwenden. Weitere Informationen zu konsistentem Hashing finden Sie unter [Konfigurieren Ihres ElastiCache-Clients für ein effizientes Load Balancing](#). Das folgende Diagramm enthält Beispiele für Memcached-Cluster mit einem Knoten und mit mehreren Knoten.

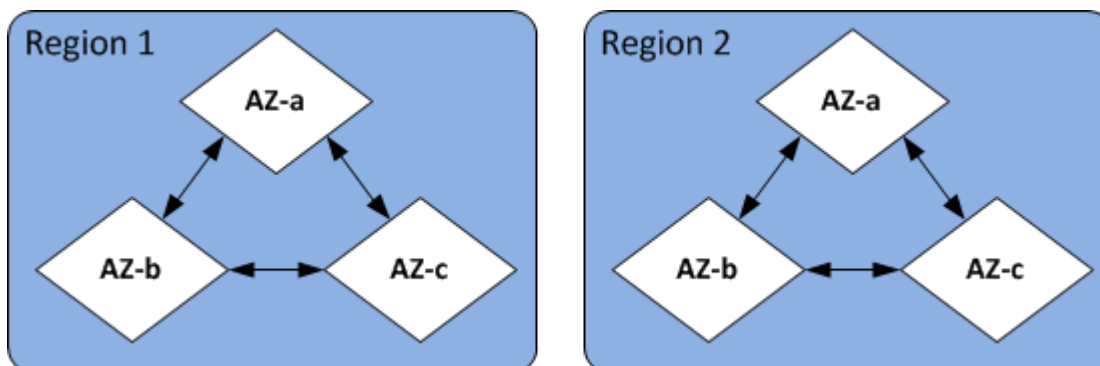


AWS Regionen und Verfügbarkeitszonen

Amazon ElastiCache for Memcached ist in mehreren AWS Regionen auf der ganzen Welt verfügbar. Somit 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 US-West (Oregon). Da die Verfügbarkeit auf neue AWS Regionen ElastiCache ausgedehnt wird, sind auch neue Endpunkte für diese AWS Regionen verfügbar, die Sie in Ihren HTTP-Anfragen, den AWS SDKs und der Konsole verwenden können. AWS CLI ElastiCache

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](#).



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

ElastiCache (Memcached) Endpunkte

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

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

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

ElastiCache Parametergruppen

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

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

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

ElastiCache Sicherheit

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

Standardmäßig werden alle neuen ElastiCache Cluster 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. Wenn Sie Ihren Cluster außerhalb von Amazon VPC ausführen möchten, können Sie Sicherheitsgruppen erstellen, um Amazon-EC2-Instances zu autorisieren, die innerhalb bestimmter Amazon-EC2-Sicherheitsgruppen ausgeführt werden.

ElastiCache Subnetzgruppen

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

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 unter [Amazon VPCs und ElastiCache Sicherheit](#), [Autorisierung des Zugriffs](#) und [Subnetze und Subnetzgruppen](#).

ElastiCache (Memcached) Ereignisse

Wenn wichtige Ereignisse in einem Cache-Cluster auftreten, ElastiCache sendet eine Benachrichtigung an ein bestimmtes Amazon SNS SNS-Thema. Zu den signifikanten Ereignissen zählen beispielsweise das Fehlschlagen des Hinzufügen eines Knotens, das erfolgreiche Hinzufügen eines Knotens und die Änderung einer Sicherheitsgruppe. Durch Überwachung auf wichtige Ereignisse kennen Sie den aktuellen Status der Cluster und können ereignisabhängig geeignete Korrekturmaßnahmen ergreifen.

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

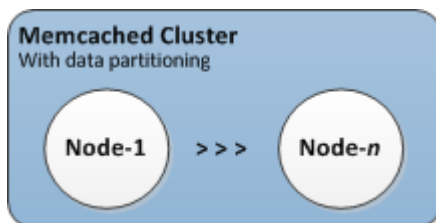
Verwalten von Clustern

Ein Cluster ist eine Sammlung einer oder mehrerer Cache-Knoten, die alle eine Instance der unterstützten Cache-Engine-Software Memcached oder 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 Memcached-Cluster. Memcached-Cluster enthalten 1 bis 60 Knoten, über die Sie Ihre Daten horizontal partitionieren.

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

Ein typischer Memcached-Cluster sieht wie folgt aus.



Die meisten ElastiCache Operationen werden auf Clusterebene ausgeführt. Sie können einen Cluster mit einer bestimmten Anzahl von Knoten und einer Parametergruppe einrichten, die die Eigenschaften für jeden Knoten steuert. Alle Knoten innerhalb eines Clusters gehören demselben Knotentyp an und verfügen über die gleichen Einstellungen für Parameter und Sicherheitsgruppen.

Jeder Cluster muss über eine Cluster-Kennung verfügen. Die Cluster-Kennung ist ein vom Kunden angegebener Name für den Cluster. Diese Kennung gibt bei der Interaktion mit der ElastiCache API und den AWS CLI Befehlen einen bestimmten Cluster an. Die Cluster-ID muss für diesen Kunden in einer AWS Region eindeutig sein.

ElastiCache unterstützt mehrere Engine-Versionen. Sofern es keine bestimmten Gründe dagegen gibt, raten wir dazu, die neueste Version zu verwenden.

ElastiCache Cluster sind für den Zugriff über eine Amazon EC2 EC2-Instance 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 [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#).

Eine Liste der unterstützten Memcached-Versionen finden Sie unter [Supported ElastiCache for Memcached Versions](#).

Auswählen eines Netzwerktyps

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er Folgendes akzeptiert:

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

IPv6 wird für Workloads unterstützt, die die Memcached-Engine ab Version 1.6.6 auf allen Instances verwenden, die auf dem [Nitro-System](#) erstellt wurden. Für den Zugriff 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, 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 betrieben zu werden, und ein reines IPv6-Subnetz, um nur als IPv6 betrieben werden zu können.

Verwenden von Dual-Stack

Wenn Sie einen Cache-Cluster erstellen und Dual-Stack als Netzwerktyp wählen, müssen Sie anschließend einen IP-Erkennungstyp festlegen — entweder IPv4 oder IPv6. ElastiCache verwendet standardmäßig den Netzwerktyp und die IP-Erkennung auf IPv6, aber das kann geändert werden. Wenn Sie Auto Discovery verwenden, werden nur die IP-Adressen des von Ihnen gewählten IP-Typs an den Memcached-Client zurückgegeben.

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 dank Auto Discovery dennoch von Vorteil, da es Migrationen (oder Rollbacks) von einem IPv4- zu einem IPv6-Discovery-IP-Typ ohne Ausfallzeiten ermöglicht.

TLS-fähige Dual-Stack-Cluster ElastiCache

Wenn TLS für ElastiCache Cluster aktiviert ist, geben die Cluster-Erkennungsfunktionen Hostnamen statt IPs `config.get_cluster` 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.

Beispiele zur Konfiguration einer IP-Protokollpräferenz bei der Auflösung von DNS-Hostnamen finden Sie unter [TLS-fähige Dual-Stack-Cluster ElastiCache](#).

Mit dem AWS Management Console

Wählen Sie beim Erstellen eines Cache-Clusters mit dem AWS Management Console unter Konnektivität 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 eines Memcached-Clusters \(Konsole\)](#).

Verwenden der -CLI

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

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id "cluster-test" \  
  --engine memcached \  
  --cache-node-type cache.m5.large \  
  --num-cache-nodes 1 \  
  --network-type dual_stack \  
  --ip-discovery-type ipv6
```

```
--ip-discovery ipv4
```

Für Windows:

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

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. Data Tiering bietet eine neue Preis-Leistungs-Option für Redis OSS-Workloads, indem zusätzlich zur Speicherung von Daten im Arbeitsspeicher 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, ElastiCache 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 OSS-Befehlen und Datenstrukturen, die in unterstützt werden. ElastiCache Um diese Funktion nutzen zu können, sind keine clientseitigen Änderungen erforderlich.

Automatische Erkennung von Knoten in Ihrem Cluster

Für Cluster, die mit der Memcached-Engine arbeiten, unterstützt ElastiCache Auto Discovery – die Fähigkeit von Client-Programmen, automatisch alle Knoten in einem Cache-Cluster zu identifizieren und Verbindungen zu all diesen Knoten zu initiieren und zu unterhalten.

Note

Auto Discovery wurde für Cache-Cluster hinzugefügt, die auf Amazon-ElastiCache-Memcached laufen.

Mit Auto Discovery muss Ihre Anwendung keine manuelle Verbindung mit einzelnen Cache-Knoten herstellen. Stattdessen stellt sie eine Verbindung mit einem Memcached-Knoten her und ruft die Liste der Knoten ab. Anhand dieser Liste erhält Ihre Anwendung Informationen über die restlichen Knoten im Cluster und kann eine Verbindung mit einem beliebigen Knoten herstellen. Die einzelnen Endpunkte der Cache-Knoten in Ihrer Anwendung müssen nicht als vordefinierter Code aufgenommen werden.

Wenn Sie den Dual-Stack-Netzwerktyp in Ihrem Cluster verwenden, gibt Auto Discovery nur IPv4- oder IPv6-Adressen zurück, je nachdem, welche Sie auswählen. Weitere Informationen finden Sie unter [Auswählen eines Netzwerktyps](#).

Alle Cache-Knoten im Cluster führen eine Liste mit Metadaten über alle anderen Knoten. Diese Metadaten werden aktualisiert, sobald Knoten hinzugefügt oder aus dem Cluster entfernt werden.

Themen

- [Vorteile von Auto Discovery](#)
- [Funktionsweise von Auto Discovery](#)

- [Verwenden von Auto Discovery](#)
- [Manuelles Herstellen von Verbindungen mit Cache-Knoten](#)
- [Hinzufügen von Auto Discovery zu Ihrer Client-Bibliothek](#)
- [ElastiCache Clients mit automatischer Erkennung](#)

Vorteile von Auto Discovery

Auto Discovery bietet die folgenden Vorteile:

- Wenn Sie die Zahl der Knoten in einem Cache-Cluster erhöhen, registrieren sich die neuen Knoten mit dem Konfigurationsendpunkt und mit allen anderen Knoten selbst. Wenn Sie Knoten aus dem Cache-Cluster entfernen, heben die entsprechenden Cluster ihre Registrierung selbst auf. In beiden Fällen werden alle anderen Knoten im Cluster mit den neuen Cache-Knotenmetadaten aktualisiert.
- Ausfälle von Cache-Knoten werden automatisch erkannt und ausgefallene Knoten automatisch ersetzt.

Note

Der Knoten fällt solange aus, bis er ersetzt wird.

- Ein Client-Programm muss nur eine Verbindung mit dem Konfigurationsendpunkt herstellen. Anschließend stellt die Auto Discovery-Bibliothek eine Verbindung mit allen anderen Knoten im Cluster her.
- Client-Programme fragen den Cluster einmal pro Minute ab (dieses Intervall kann bei Bedarf geändert werden). Wenn Änderungen an der Cluster-Konfiguration vorgenommen wurden, z. B. neue oder gelöschte Knoten, erhält der Client eine aktualisierte Liste mit Metadaten. Anschließend stellt der Client je nach Bedarf eine Verbindung mit diesen Knoten her bzw. trennt die Verbindung.

Die Auto-Discovery-Funktion ist in allen ElastiCache-Memcached-Cache-Clustern aktiviert. Sie müssen Ihre Cache-Knoten nicht neu starten, um diese Funktion zu verwenden.

Funktionsweise von Auto Discovery

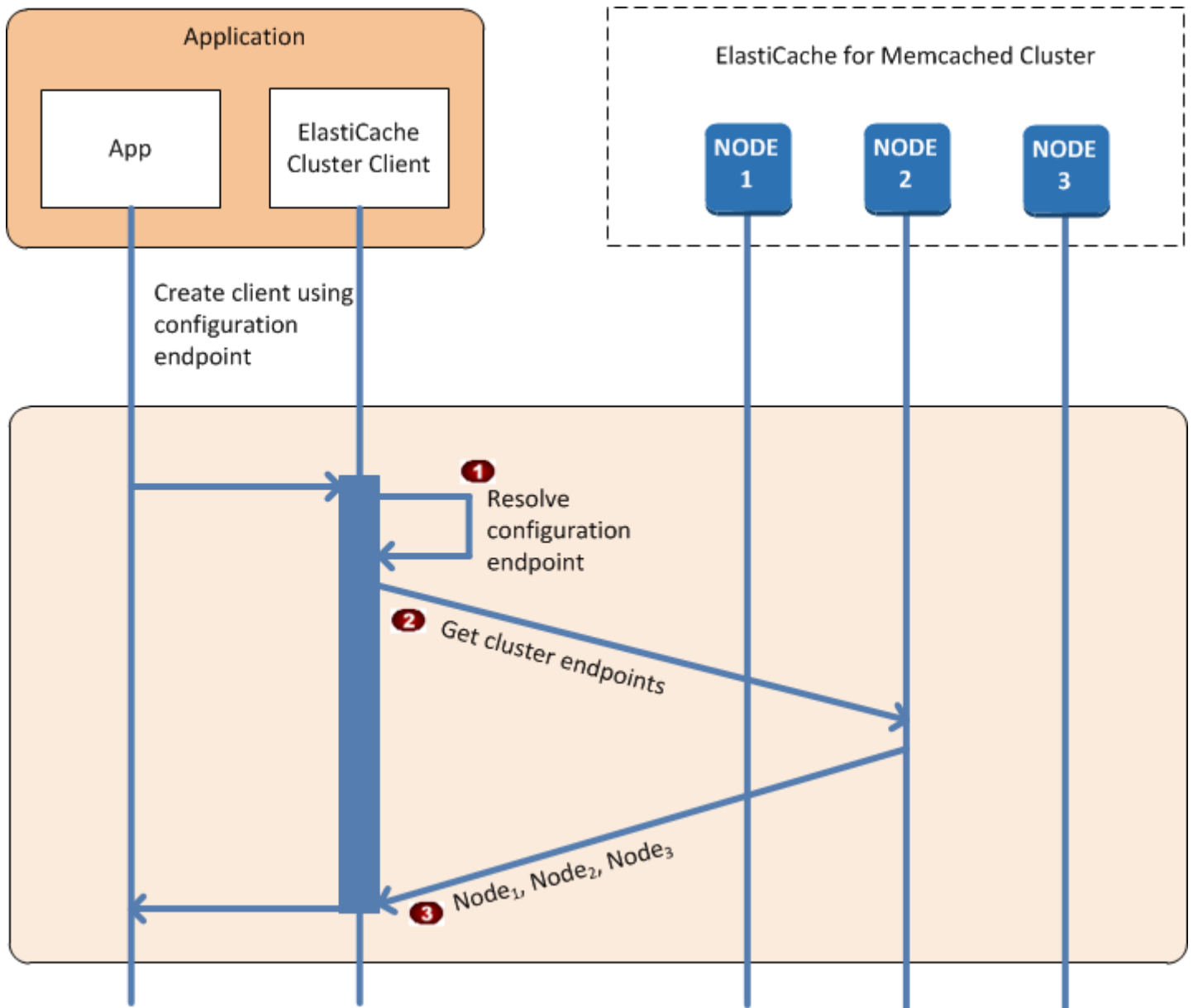
Themen

- [Herstellen von Verbindungen mit Cache-Knoten](#)
- [Normale Cluster-Operationen](#)
- [Weitere Operationen in](#)

Dieser Abschnitt beschreibt, wie Client-Anwendungen ElastiCache-Cluster-Client verwenden, um Cache-Knoten-Verbindungen zu verwalten und mit Datenelementen im Cache zu interagieren.

Herstellen von Verbindungen mit Cache-Knoten

Von der Anwendungsseite aus betrachtet, unterscheidet sich das Herstellen einer Verbindung mit dem Cluster-Konfigurationsendpunkt nicht vom Herstellen einer direkten Verbindung mit einem einzelnen Cache-Knoten. Das folgende Ablaufdiagramm zeigt den Prozess zum Herstellen der Verbindung mit Cache-Knoten.



Prozess zum Herstellen von Verbindungen mit Cache-Knoten

- Die Anwendung löst den DNS-Namen des Konfigurationsendpunkts auf. Da der Konfigurationsendpunkt CNAME-Einträge für alle Cache-Knoten verwaltet, wird der DNS-Name in einen der Knoten aufgelöst. Der Client kann dann eine Verbindung mit diesem Knoten herstellen.
- Der Client fordert die Konfigurationsinformationen für alle anderen Knoten an. Da jeder Knoten Konfigurationsinformationen für alle Knoten im Cluster verwaltet, kann jeder Knoten auf Anfrage Konfigurationsinformationen an den Client weitergeben.

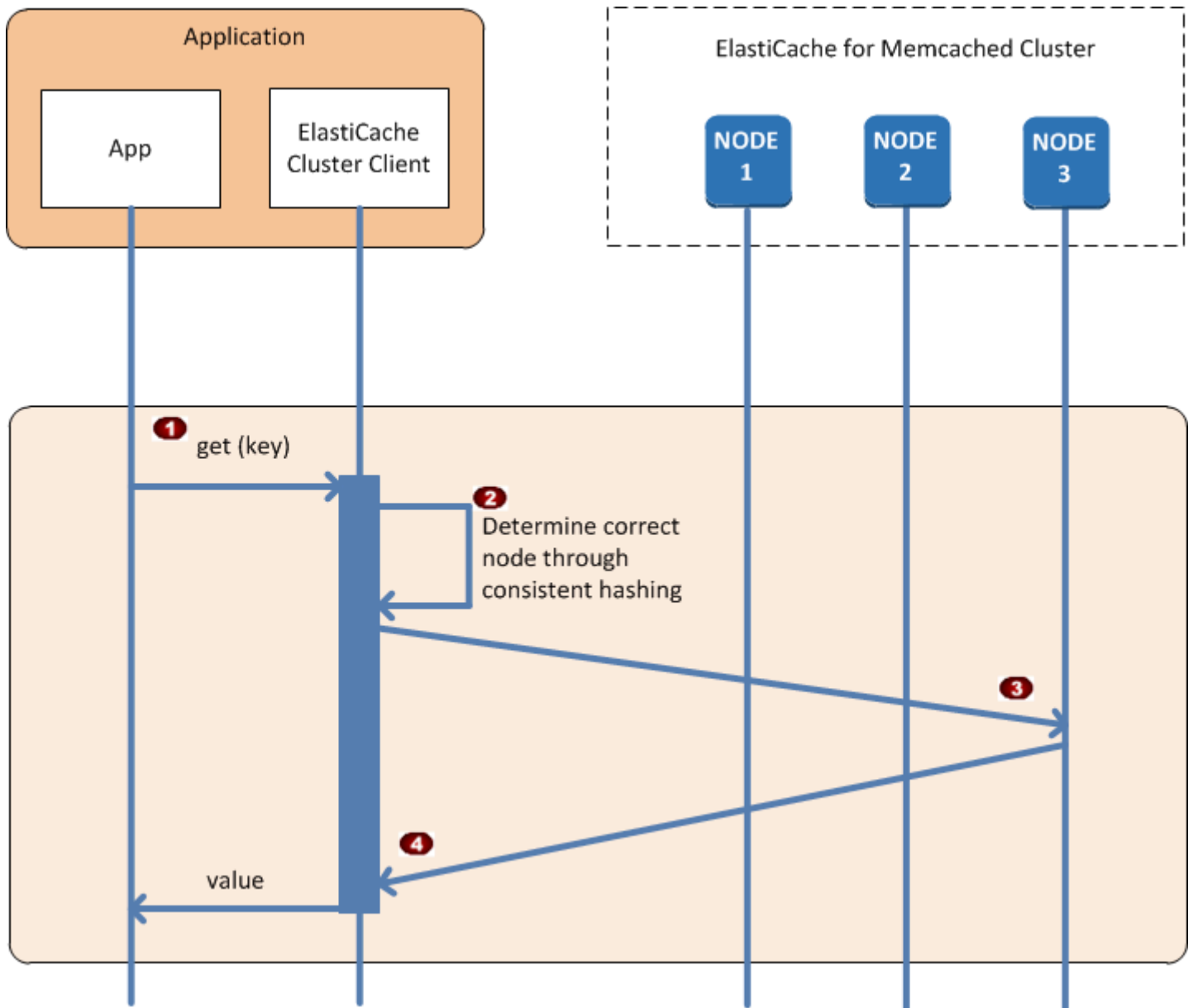
- Der Client empfängt die aktuelle Liste mit den Hostnamen und IP-Adressen der Cache-Knoten. Anschließend kann er eine Verbindung mit allen anderen Knoten im Cluster herstellen.

Note

Das Client-Programm aktualisiert seine Liste mit den Hostnamen und IP-Adressen der Cache-Knoten einmal pro Minute. Dieses Abfrageintervall kann bei Bedarf angepasst werden.

Normale Cluster-Operationen

Wenn die Anwendung eine Verbindung zu allen Cache-Knoten hergestellt hat, bestimmt der ElastiCache-Cluster-Client, welche Knoten einzelne Datenelemente speichern sollen und welche Knoten später nach diesen Datenelementen abgefragt werden sollen. Das folgende Ablaufdiagramm zeigt den Prozess für normale Cluster-Operationen.



Prozess für normale Cluster-Operationen

- Die Anwendung erstellt eine `get`-Anforderung für ein bestimmtes Datenelement, das durch seinen Schlüssel identifiziert wird.
- Der Client verwendet einen Hashing-Algorithmus für den Schlüssel, um den Cache-Knoten zu ermitteln, der das Datenelement enthält.
- Das Datenelement wird vom entsprechenden Knoten angefordert.
- Das Datenelement wird an die Anwendung zurückgegeben.

Weitere Operationen in

In manchen Situationen kann es vorkommen, dass Sie eine Änderung an den Knoten eines Clusters vornehmen. So können Sie beispielsweise einen zusätzlichen Knoten hinzufügen, um zusätzlichen Bedarf zu decken, oder einen Knoten löschen, um in Zeiten geringerer Nachfrage Geld zu sparen. Oder Sie ersetzen einen Knoten, weil er auf die eine oder andere Weise ausgefallen ist.

Bei einer Änderung im Cluster, die eine Aktualisierung der Metadaten auf die Endpunkte des Clusters erfordert, wird diese Änderung auf allen Knoten gleichzeitig vorgenommen. So sind die Metadaten in einem bestimmten Knoten konsistent mit den Metadaten in allen anderen Knoten im Cluster.

In jedem Fall sind die Metadaten für alle Knoten jederzeit konsistent, da sie für alle Knoten im Cluster gleichzeitig aktualisiert werden. Verwenden Sie immer den Konfigurationsendpunkt, um die Endpunkte der verschiedenen Knoten im Cluster zu erhalten. Durch Verwenden des Konfigurationsendpunkts stellen Sie sicher, dass Sie keine Endpunktdaten von einem Knoten erhalten, der "verschwindet".

Hinzufügen eines Knotens

Während der Zeit, in der der Knoten eingerichtet wird, ist sein Endpunkt nicht in den Metadaten enthalten. Sobald der Knoten verfügbar ist, wird er den jeweiligen Metadaten der Cluster-Knoten hinzugefügt. In diesem Szenario sind die Metadaten für alle Knoten konsistent und Sie können mit dem neuen Knoten erst interagieren, sobald er verfügbar ist. Vorher liegen Ihnen keine Informationen darüber vor und Sie interagieren mit den Knoten in Ihrem Cluster so, als ob der neue Knoten nicht vorhanden wäre.

Löschen eines Knotens

Wenn ein Knoten entfernt wird, wird sein Endpunkt erst in den Metadaten gelöscht und anschließend wird der Knoten aus dem Cluster entfernt. In diesem Szenario sind die Metadaten in allen Knoten konsistent und zu keiner Zeit enthalten sie den Endpunkt für den Knoten, der entfernt werden soll, während der Knoten nicht verfügbar ist. Während des Zeitraums, in dem der Knoten entfernt wird, erfolgt keine Meldung in den Metadaten. Demzufolge interagiert Ihre Anwendung nur mit den verbleibenden $n-1$ Knoten, so als ob der Knoten nicht vorhanden wäre.

Ersetzen eines Knotens

Wenn ein Knoten ausfällt, schaltet ElastiCache diesen Knoten ab und fährt einen Ersatz hoch. Das Ersetzen dauert einige Minuten. Während dieser Zeit zeigen die Metadaten in allen Knoten

den Endpunkt für den ausgefallenen Knoten noch an, doch jeder Versuch, mit dem Knoten zu interagieren, schlägt fehl. Daher sollte Ihre Logik immer Logik für Wiederholungsversuche umfassen.

Verwenden von Auto Discovery

Zur Verwendung von Auto Discovery führen Sie die folgenden Schritte aus:

- [Schritt 1: Abrufen des Konfigurationsendpunkts](#)
- [Schritt 2: Herunterladen des ElastiCache-Cluster-Clients](#)
- [Schritt 3: Ändern Ihres Anwendungsprogramms](#)

Schritt 1: Abrufen des Konfigurationsendpunkts

Zum Herstellen der Verbindung mit einem Cluster müssen die Client-Programme den Cluster-Konfigurationsendpunkt kennen. Weitere Informationen finden Sie im Thema [Finden der Endpunkte eines Clusters \(Konsole\)](#).

Sie können auch den Befehl `aws elasticache describe-cache-clusters` mit dem Parameter `--show-cache-node-info` verwenden:

Unabhängig von der Methode, mit der Sie die Endpunkte des Clusters suchen, enthält der Konfigurationsendpunkt in seiner Adresse immer `.cfg`.

Example Suchen von Endpunkten mit der AWS CLI für ElastiCache

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

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

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  

```

```
{
  "CacheNodeId": "0001",
  "Endpoint": {
    "Port": 11211,
    "Address": "mycluster.fnjyzo.cfg.0001.use1.cache.amazonaws.com"
  },
  "CacheNodeStatus": "available",
  "ParameterGroupStatus": "in-sync",
  "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
  "CustomerAvailabilityZone": "us-east-1e"
},
{
  "CacheNodeId": "0002",
  "Endpoint": {
    "Port": 11211,
    "Address": "mycluster.fnjyzo.cfg.0002.use1.cache.amazonaws.com"
  },
  "CacheNodeStatus": "available",
  "ParameterGroupStatus": "in-sync",
  "CacheNodeCreateTime": "2016-10-12T21:39:28.001Z",
  "CustomerAvailabilityZone": "us-east-1a"
}
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "Multiple",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-10-12T21:39:28.001Z",
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "available",
"NumCacheNodes": 2,
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
"CacheSubnetGroupName": "default",
"EngineVersion": "1.4.24",
"PendingModifiedValues": {},
```

```
        "PreferredMaintenanceWindow": "sat:06:00-sat:07:00",
        "CacheNodeType": "cache.r3.large"
    }
]
}
```

Schritt 2: Herunterladen des ElastiCache-Cluster-Clients

Um die Vorteile der automatischen Erkennung nutzen zu können, müssen Client-Programme den ElastiCache-Cluster-Client verwenden. Der ElastiCache-Cluster-Client ist für Java, PHP und .NET verfügbar und enthält die gesamte notwendige Logik für die Erkennung und Verbindung zu allen Cache-Knoten.

So laden Sie den ElastiCache-Cluster-Client herunter

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie in der ElastiCache-Konsole ElastiCache-Cluster-Client und dann Herunterladen.

Der Quellcode für den ElastiCache-Cluster Client für Java ist unter <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java> verfügbar. Diese Bibliothek basiert auf dem häufig verwendeten Spymemcached-Client. Der ElastiCache-Cluster-Client wird unter der Amazon-Software-Lizenz <https://aws.amazon.com/asl> veröffentlicht. Es steht Ihnen frei, den Quellcode nach eigenen Wünschen zu ändern. Sie können den Code sogar in andere Memcached-Open-Source-Bibliotheken oder in Ihren eigenen Client-Code integrieren.

Note

Um den ElastiCache-Cluster-Client für PHP zu verwenden, müssen Sie ihn zunächst auf Ihrer Amazon-EC2-Instance installieren. Weitere Informationen finden Sie unter [Installieren des ElastiCache-Cluster-Clients-für-PHP](#).

Für einen von TLS unterstützten Client laden Sie die Binärdatei mit PHP-Version 7.4 oder höher herunter.

Um den ElastiCache-Cluster-Client für .NET zu verwenden, müssen Sie ihn zunächst auf Ihrer Amazon-EC2-Instance installieren. Weitere Informationen finden Sie unter [Installieren des ElastiCache-Cluster-Clients für .NET](#).

Schritt 3: Ändern Ihres Anwendungsprogramms

Ändern Sie Ihr Anwendungsprogramm, sodass es Auto Discovery verwendet. Die folgenden Abschnitte zeigen, wie Sie den ElastiCache-Cluster-Client für Java, PHP und .NET verwenden.

Important

Stellen Sie beim Angeben des Cluster-Konfigurationsendpunkts sicher, dass der Endpunkt in seiner Adresse ".cfg" enthält, wie hier dargestellt. Verwenden Sie keinen CNAME oder einen Endpunkt ohne ".cfg" in der Adresse.

```
"mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
```

Wenn Sie den Cluster-Konfigurationsendpunkt nicht explizit angeben, erfolgt die Konfiguration für einen spezifischen Knoten.

Verwenden des ElastiCache-Cluster-Clients für Java

Das folgende Programm zeigt, wie der ElastiCache-Cluster-Client verwendet wird, um eine Verbindung zu einem Cluster-Konfigurationsendpunkt herzustellen und ein Datenelement zum Cache hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

```
package com.amazon.elasticache;

import java.io.IOException;
import java.net.InetSocketAddress;

// Import the &AWS;-provided library with Auto Discovery support
import net.spy.memcached.MemcachedClient;

public class AutoDiscoveryDemo {

    public static void main(String[] args) throws IOException {

        String configEndpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";
        Integer clusterPort = 11211;

        MemcachedClient client = new MemcachedClient(
            new InetSocketAddress(configEndpoint,
```

```
        clusterPort));  
    // The client will connect to the other cache nodes automatically.  
  
    // Store a data item for an hour.  
    // The client will decide which cache host will store this item.  
    client.set("theKey", 3600, "This is the data value");  
} } }
```

Verwenden des ElastiCache-Cluster-Clients für PHP

Das folgende Programm zeigt, wie der ElastiCache-Cluster-Client verwendet wird, um eine Verbindung zu einem Cluster-Konfigurationsendpunkt herzustellen und ein Datenelement zum Cache hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

Um den ElastiCache-Cluster-Client für PHP zu verwenden, müssen Sie ihn zunächst auf Ihrer Amazon-EC2-Instance installieren. Weitere Informationen finden Sie unter [Installieren des ElastiCache-Cluster-Clients-für-PHP](#)

```
<?php  
  
/**  
 * Sample PHP code to show how to integrate with the Amazon ElastiCache  
 * Auto Discovery feature.  
 */  
  
/* Configuration endpoint to use to initialize memcached client.  
 * This is only an example. */  
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";  
  
/* Port for connecting to the ElastiCache cluster.  
 * This is only an example */  
$server_port = 11211;  
  
/**  
 * The following will initialize a Memcached client to utilize the Auto Discovery  
 * feature.  
 *  
 * By configuring the client with the Dynamic client mode with single endpoint, the  
 * client will periodically use the configuration endpoint to retrieve the current  
 * cache
```



```
* cluster configuration. This allows scaling the cache cluster up or down in number
of nodes
* without requiring any changes to the PHP application.
*
* By default the Memcached instances are destroyed at the end of the request.
* To create an instance that persists between requests,
*   use persistent_id to specify a unique ID for the instance.
* All instances created with the same persistent_id will share the same connection.
* See http://php.net/manual/en/memcached.construct.php for more information.
*/
$dynamic_client = new Memcached('persistent-id');
$dynamic_client->setOption(Memcached::OPT_CLIENT_MODE,
Memcached::DYNAMIC_CLIENT_MODE);
$dynamic_client->addServer($server_endpoint, $server_port);

/**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
 */
$dynamic_client->set('key', 'value', 60);

/**
 * Configuring the client with Static client mode disables the usage of Auto Discovery
 * and the client operates as it did before the introduction of Auto Discovery.
 * The user can then add a list of server endpoints.
 */
$static_client = new Memcached('persistent-id');
$static_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::STATIC_CLIENT_MODE);
$static_client->addServer($server_endpoint, $server_port);

/**
 * Store the data without expiration.
 * The client will decide which cache host will store this item.
 */
$static_client->set('key', 'value');
?>
```

Ein Beispiel zur Verwendung des ElastiCache-Cluster-Clients mit aktiviertem TLS finden Sie unter [Verwendung der Verschlüsselung während der Übertragung mit PHP und Memcached](#).

Verwendung des ElastiCache-Cluster-Clients für .NET

Note

Der ElastiCache-.NET-Clusterclient ist seit Mai 2022 veraltet.

.NET-Client für ElastiCache ist Open Source unter <https://github.com/awslabs/elasticache-cluster-config-net>.

Die .NET-Anwendungen erhalten ihre Konfigurationen in der Regel von ihrer Config-Datei. Nachfolgend finden Sie ein Beispiel einer Anwendungs-Config-Datei.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section
      name="clusterclient"
      type="Amazon.ElastiCacheCluster.ClusterConfigSettings,
Amazon.ElastiCacheCluster" />
  </configSections>

  <clusterclient>
    <!-- the hostname and port values are from step 1 above -->
    <endpoint hostname="mycluster.fnjyzo.cfg.use1.cache.amazonaws.com"
port="11211" />
  </clusterclient>
</configuration>
```

Das folgende C#-Programm zeigt, wie der ElastiCache-Cluster-Client verwendet wird, um eine Verbindung zu einem Cluster-Konfigurationsendpunkt herzustellen und ein Datenelement zum Cache hinzuzufügen. Mithilfe von Auto Discovery stellt das Programm eine Verbindung mit allen Knoten im Cluster her, ohne dass ein Eingreifen erforderlich ist.

```
// *****
// Sample C# code to show how to integrate with the Amazon ElastiCache Auto Discovery
// feature.

using System;

using Amazon.ElastiCacheCluster;
```

```
using Enyim.Caching;
using Enyim.Caching.Memcached;

public class DotNetAutoDiscoveryDemo {

    public static void Main(String[] args) {

        // instantiate a new client.
        ElastiCacheClusterConfig config = new ElastiCacheClusterConfig();
        MemcachedClient memClient = new MemcachedClient(config);

        // Store the data for 3600 seconds (1hour) in the cluster.
        // The client will decide which cache host will store this item.
        memClient.Store(StoreMode.Set, 3600, "This is the data value.");

    } // end Main

} // end class DotNetAutoDiscoverDemo
```

Manuelles Herstellen von Verbindungen mit Cache-Knoten

Wenn Ihr Client-Programm die Auto Discovery-Funktion nicht verwendet, kann es die Verbindung mit den einzelnen Cache-Knoten manuell herstellen. Dies ist das Standardverhalten für Memcached-Clients.

Eine Liste der Hostnamen und Portnummern der Cache-Knoten erhalten Sie über die [AWS Managementkonsole](#). Sie können auch den Befehl AWS CLI `aws elasticache describe-cache-clusters` mit dem Parameter `--show-cache-node-info` verwenden:

Example

Der folgende Java-Codeausschnitt zeigt, wie eine Verbindung mit allen Knoten in einem aus vier Knoten bestehenden Cache-Cluster hergestellt wird:

```
...  
  
ArrayList<String> cacheNodes = new ArrayList<String>(  
    Arrays.asList(  
        "mycachecluster.fnjyzo.0001.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0002.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0003.use1.cache.amazonaws.com:11211",  
        "mycachecluster.fnjyzo.0004.use1.cache.amazonaws.com:11211"));  
  
MemcachedClient cache = new MemcachedClient(AddrUtil.getAddresses(cacheNodes));  
  
...
```

Important

Wenn Sie Ihren Cache-Cluster durch Hinzufügen oder Entfernen von Knoten nach oben oder nach unten skalieren, müssen Sie die Liste der Knoten im Client-Code aktualisieren.

Hinzufügen von Auto Discovery zu Ihrer Client-Bibliothek

Die Konfigurationsinformationen für Auto Discovery werden in jedem Cache-Cluster-Knoten redundant gespeichert. Client-Anwendungen können jeden Cache-Knoten abfragen und die Konfigurationsinformationen für alle Knoten im Cluster abrufen.

Die jeweilige Methode, die eine Anwendung dazu nutzt, hängt von der Cache-Engine-Version ab:

- Wenn die Cache-Engine-Version 1.4.14 oder höher ist, verwenden Sie den Befehl `config`.
- Wenn die Cache-Engine-Version älter als 1.4.14 ist, verwenden Sie den Befehl `get AmazonElastiCache:cluster`.

Die Ausgaben dieser beiden Befehle sind identisch und werden im Abschnitt [Ausgabeformat](#) beschrieben.

Cache-Engine-Version 1.4.14 oder höher

Für die Cache-Engine-Version 1.4.14 oder höher verwenden Sie den Befehl `config`. Dieser Befehl wurde von ElastiCache zu den Memcached-ASCII- und -Binärprotokollen hinzugefügt und ist im ElastiCache-Cluster-Client implementiert. Wenn Sie Auto Discovery mit einer anderen Client-Bibliothek verwenden möchten, muss diese Bibliothek um den Support für den Befehl `config` erweitert werden.

Note

Die folgende Dokumentation bezieht sich zwar auf das ASCII-Protokoll, der Befehl `config` unterstützt jedoch sowohl das ASCII- als auch das Binärprotokoll. Wenn Sie Auto-Discovery-Support unter Verwendung des binären Protokolls hinzufügen möchten, lesen Sie den [Quellcode für den ElastiCache-Cluster-Client](#).

Syntax

```
config [sub-command] [key]
```

Optionen

Name	Beschreibung	Erforderlich
sub-command	Der Unterbefehl, der zum Interagieren mit einem Cache-Knoten verwendet wird. Für Auto Discovery lautet dieser Unterbefehl <code>get</code> .	Ja
key	Der Schlüssel, mit dem die Cluster-Konfiguration gespeichert ist. Für Auto Discovery lautet dieser Schlüssel <code>cluster</code> .	Ja

Verwenden Sie zum Abrufen der Cluster-Konfigurationsinformationen den folgenden Befehl:

```
config get cluster
```

Cache-Engine-Version älter als 1.4.14

Verwenden Sie zum Abrufen der Cluster-Konfigurationsinformationen den folgenden Befehl:

```
get AmazonElastiCache:cluster
```

Note

Nehmen Sie keine unerlaubten Änderungen am "AmazonElastiCache:cluster"-Schlüssel vor, da hiermit die Cluster-Konfigurationsinformationen gespeichert sind. Wenn Sie diesen Schlüssel überschreiben, kann der Client für eine kurze Zeit (nicht mehr als 15 Sekunden) falsch konfiguriert sein, bevor ElastiCache die Konfigurationsinformationen automatisch und korrekt aktualisiert.

Ausgabeformat

Wenn Sie `config get cluster` oder `get AmazonElastiCache:cluster` verwenden, umfasst die Antwort zwei Zeilen:

- Die Versionsnummer der Konfigurationsinformationen. Jedes Mal, wenn ein Knoten hinzugefügt oder aus dem Cache-Cluster entfernt wird, wird die Versionsnummer um eins erhöht.

- Eine Liste der Cache-Knoten. Jeder Knoten in der Liste wird durch eine `hostname|ip-address|port`-Gruppe repräsentiert und durch ein Leerzeichen getrennt.

Am Ende jeder Zeile erscheint ein Wagenrücklauf- und Zeilenvorschubzeichen (CR + LF). Die Datenzeile enthält am Ende ein Zeilenvorschubzeichen (LF), dem die Zeichen CR + LF hinzugefügt werden. Die Config-Versionszeile wird durch LF ohne CR beendet.

Ein Cache-Cluster mit drei Knoten wird wie folgt dargestellt:

```
configversion\n
hostname|ip-address|port hostname|ip-address|port hostname|ip-address|port\n\r\n
```

Jeder Knoten wird sowohl mit dem CNAME als auch der privaten IP-Adresse angezeigt. Der CNAME ist immer vorhanden. Wenn die private IP-Adresse nicht verfügbar ist, wird sie nicht angezeigt. Die Pipe-Zeichen "|" werden dennoch gedruckt.

Example

Das nachfolgende Beispiel zeigt die Nutzlast, die zurückgegeben wird, wenn Sie die Konfigurationsinformationen abfragen:

```
CONFIG cluster 0 136\r\n
12\n
myCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211
  myCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\n\r\n
END\r\n
```

Note

- Die zweite Zeile gibt an, dass die Konfigurationsinformationen bisher zwölfmal geändert wurden.
- In der dritten Zeile wird die Liste der Knoten in alphabetischer Reihenfolge nach Hostname angegeben. Diese Reihenfolge kann sich von der Sequenz unterscheiden, die Sie derzeit in Ihrer Client-Anwendung verwenden.

ElastiCache Clients mit automatischer Erkennung

Dieser Abschnitt beschreibt die Installation und Konfiguration der ElastiCache PHP- und .NET-Clients.

Themen

- [Installieren und Kompilieren von Cluster-Clients](#)
- [Konfiguration von ElastiCache Clients](#)

Installieren und Kompilieren von Cluster-Clients

Dieser Abschnitt behandelt die Installation, Konfiguration und Kompilierung der PHP- und .NET-Clients des Amazon-ElastiCache-Auto-Discovery-Clusters.

Themen

- [Installieren des ElastiCache-Cluster-Clients für .NET](#)
- [Installieren des ElastiCache-Cluster-Clients-für-PHP](#)
- [Kompilieren des Quellcodes für den ElastiCache-Cluster-Client für PHP](#)

Installieren des ElastiCache-Cluster-Clients für .NET

Note

Der ElastiCache-.NET-Clusterclient ist seit Mai 2022 veraltet.

Sie können den ElastiCache-.NET-Cluster-Client-Code als Open Source unter <https://github.com/awslabs/elasticache-cluster-config-net> finden.

In diesem Abschnitt wird beschrieben, wie Sie die .NET-Komponenten für den ElastiCache-Cluster-Client auf Amazon-EC2-Instances installieren, aktualisieren und entfernen. Weitere Informationen zur automatischen Erkennung finden Sie unter [Auto Discovery](#). .NET-Beispielcode für die Verwendung des Clients finden Sie unter [Auto Discovery mit DotNET](#).

Themen

- [Installieren von .NET](#)
- [Herunterladen des .NET-Cluster-Clients für ElastiCache](#)

- [AWS-Komponenten mit NuGet installieren](#)

Installieren von .NET

Sie müssen .NET 3.5 oder höher installiert haben, um das AWS-.NET-SDK für ElastiCache zu verwenden. Wenn .NET 3.5 oder höher nicht installiert ist, können Sie die aktuelle Version unter <http://www.microsoft.com/net> herunterladen, um sie zu installieren.

Herunterladen des .NET-Cluster-Clients für ElastiCache

Zum Herunterladen des .NET-Cluster-Clients für ElastiCache

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf ElastiCache Cluster Client.
3. Wählen Sie in der Liste Download ElastiCache Memcached Cluster Client den Eintrag .NET aus und klicken Sie auf Download.

AWS-Komponenten mit NuGet installieren

NuGet ist ein Paketverwaltungssystem für die .NET-Plattform. NuGet berücksichtigt die Abhängigkeiten der Komponenten und installiert alle erforderlichen Dateien automatisch. Die von NuGet installierten Komponenten werden statt an einem zentralen Ort wie Program Files mit Ihrer Lösung gespeichert, sodass Sie anwendungsspezifische Versionen installieren können, ohne dass Kompatibilitätsprobleme auftreten.

Installieren von NuGet

NuGet kann von der Installation Gallery auf MSDN unter <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c> installiert werden. Wenn Sie Visual Studio 2010 oder höher verwenden wird NuGet automatisch installiert.

Sie können NuGet entweder mit dem Projektmappen-Explorer oder der Paket-Manager-Konsole verwenden.

Verwenden von NuGet mit dem Projektmappen-Explorer

So verwenden Sie NuGet mit dem Projektmappen-Explorer in Visual Studio 2010

1. Wählen Sie im Menü Extras den Bibliotheks-Paket-Manager aus.

2. Klicken Sie auf Paket-Manager-Konsole.

So verwenden Sie NuGet mit dem Projektmappen-Explorer in Visual Studio 2012 oder Visual Studio 2013

1. Wählen Sie im Menü Extras den NuGet-Paket-Manager aus.
2. Klicken Sie auf Paket-Manager-Konsole.

An der Befehlszeile können Sie die Komponenten mit `Install-Package` wie im Folgenden dargestellt installieren.

```
Install-Package Amazon.ElastiCacheCluster
```

Eine Seite für jedes Paket, das über NuGet verfügbar ist, z. B. AWS-SDK und AWS-.Extensions-Assemblies, finden Sie auf der NuGet-Website unter <http://www.nuget.org>. Die Seite für die einzelnen Pakete umfasst eine Beispielbefehlszeile zum Installieren des Pakets mit der Konsole und eine Liste der vorherigen Paketversionen, die über NuGet verfügbar sind.

Weitere Informationen zu den Befehlen der Paket-Manager-Konsole finden Sie unter <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>.

Installieren des ElastiCache-Cluster-Clients-für-PHP

In diesem Abschnitt wird beschrieben, wie Sie die PHP-Komponenten für den ElastiCache-Cluster-Client auf Amazon-EC2-Instances installieren, aktualisieren und entfernen. Weitere Informationen zu Auto Discovery finden Sie unter [Automatische Erkennung von Knoten in Ihrem Cluster](#). Beispiele für PHP-Code zur Verwendung des Clients finden Sie unter [Verwenden des ElastiCache-Cluster-Clients für PHP](#).

Themen

- [Herunterladen des Installationspakets](#)
- [Für Benutzer, die bereits die Erweiterung php-memcached installiert haben](#)
- [Installationsschritte für neue Benutzer](#)
- [Entfernen des PHP-Cluster-Clients](#)

Herunterladen des Installationspakets

Um sicherzustellen, dass Sie die richtige Version des ElastiCache-Cluster-Client-für-PHP verwenden, müssen Sie wissen, welche Version von PHP auf Ihrer Amazon-EC2-Instance installiert ist. Sie müssen auch wissen, ob auf Ihrer Amazon-EC2-Instance eine 64-Bit- oder 32-Bit-Version von Linux läuft.

So ermitteln Sie die auf Ihrer Amazon-EC2-Instance installierte PHP-Version

- Führen Sie in der Befehlszeile den folgenden Befehl aus:

```
php -v
```

Die PHP-Version wird wie in diesem Beispiel ausgegeben:

```
PHP 5.4.10 (cli) (built: Jan 11 2013 14:48:57)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.4.0, Copyright (c) 1998-2012 Zend Technologies
```

Note

Wenn Ihre PHP- und Memcached-Versionen inkompatibel sind, wird in etwa folgende Fehlermeldung ausgegeben:

```
PHP Warning: PHP Startup: memcached: Unable to initialize module
Module compiled with module API=20100525
PHP compiled with module API=20131226
These options need to match
in Unknown on line 0
```

In diesem Fall müssen Sie das Modul aus dem Quellcode kompilieren. Weitere Informationen finden Sie unter [Kompilieren des Quellcodes für den ElastiCache-Cluster-Client für PHP](#).

So bestimmen Sie Ihre Amazon-EC2-AMI-Architektur (64-Bit oder 32-Bit)

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. In der Instances-Liste klicken Sie auf Ihre Amazon-EC2-Instance.
3. Suchen Sie auf der Registerkarte Description nach dem Feld AMI:. Die Beschreibung einer 64-Bit-Instance enthält x86_64, bei einer 32-Bit-Instance enthält dieses Feld i386 oder i686.

Sie können nun den ElastiCache-Cluster-Client herunterladen.

So laden Sie den ElastiCache-Cluster-Client-für-PHP herunter

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 ElastiCache-Konsole ElastiCache-Cluster-Client aus.
3. Wählen Sie in der Liste ElastiCache-Memcached-Cluster-Client heruntergeladen den ElastiCache-Cluster-Client aus, der Ihrer PHP-Version und AMI-Architektur entspricht, und klicken Sie dann auf die Schaltfläche Herunterladen.

Für einen von TLS unterstützten Client laden Sie die Binärdatei mit PHP-Version 7.4 oder höher herunter.

Für Benutzer, die bereits die Erweiterung php-memcached installiert haben

So aktualisieren Sie die **php-memcached**-Installation

1. Entfernen Sie die bisherige Installation der Memcached-Erweiterung für PHP, wie im Thema [Entfernen des PHP-Cluster-Clients](#) beschrieben.
2. Installieren Sie die neue ElastiCache-Erweiterung php-memcached wie zuvor in [Installationsschritte für neue Benutzer](#) beschrieben.

Installationsschritte für neue Benutzer

Themen


- [Installieren von PHP 7.x bis 8.x für neue Benutzer](#)
- [Installieren von PHP 5.x für neue Benutzer](#)

Installieren von PHP 7.x bis 8.x für neue Benutzer

Themen

- [Installieren Sie PHP 7.x bis 8.x auf einem Amazon-Linux-2-AMI wie folgt:](#)
- [Installieren Sie PHP 7.x bis 8.x auf einem Amazon Linux 201609 AMI wie folgt:](#)
- [Installieren Sie PHP 7.x bis 8.x auf einem SUSE-Linux-15-AMI wie folgt:](#)
- [Installieren Sie PHP 7.x bis 8.x auf einem Ubuntu-22.04-AMI wie folgt:](#)

Installieren Sie PHP 7.x bis 8.x auf einem Amazon-Linux-2-AMI wie folgt:

 Note

Ersetzen Sie *PHP-7.x* wenn nötig mit der Version, die Sie verwenden.

1. Starten Sie eine neue Instance aus dem AMI.
2. Führen Sie den Befehl aus:

```
sudo yum install gcc-c++ zlib-devel
```

3. Installieren Sie PHP 7.x mit `amazon-linux-extras`

Mit Amazon Linux 2 können Sie die Extras-Bibliothek verwenden, um Anwendungs- und Software-Updates auf Ihren Instances zu installieren. Dieses Software-Updates werden als Themen bezeichnet. Sie können eine bestimmte Version eines Themas installieren oder die Versionsinformationen weglassen, um die neueste Version zu verwenden. Weitere Informationen finden Sie in der [Extras-Bibliothek \(Amazon Linux 2\)](#).

Gehen Sie dazu wie folgt vor;

- a. Überprüfen Sie zunächst, ob `amazon-linux-extras` bereits installiert ist.
- b. Falls nicht installiert, verwenden Sie den folgenden Befehl zur Installation:

```
sudo yum install -y amazon-linux-extras
```

- c. Bestätigen Sie, dass das PHP-7.x-Thema auf dem Amazon-Linux-2-Rechner verfügbar ist:

```
sudo amazon-linux-extras | grep php
```

- d. Überprüfen Sie in der Ausgabe alle PHP-7-Themen und wählen Sie die gewünschte Version aus:

```
sudo amazon-linux-extras enable php7.x
```

- e. Installieren Sie PHP-Pakete aus dem Repository. Zum Beispiel:

```
sudo yum clean metadata
```

```
sudo yum install php php-devel
```

4. Laden Sie den Amazon ElastiCache-Cluster-Client herunter.

- Öffnen Sie die ElastiCache-Konsole unter <https://console.aws.amazon.com/elasticache/>.

Gehen Sie im ElastiCache-Dashboard zu ElastiCache Cluster Client und wählen Sie die gewünschte PHP7-Version.

- Ersetzen Sie `PHP-7.X` über die Befehlszeile durch die gewünschte PHP-Version und ersetzen Sie `ARCH` durch die gewünschte Architektur (X86 oder arm) und für `PHP >= 7.4` ersetzen Sie `OpenSSL` durch die gewünschte OpenSSL-Version (`openssl1.1` oder `openssl3`). Wenn Sie `PHP > 7.4` verwenden, entfernen Sie das `OpenSSL`-Suffix.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.X/  
latest-64bit-<ARCH>-<OpenSSL>
```

5. Verwenden Sie `tar -zxvf`, um die heruntergeladene Datei zu extrahieren.

```
tar -zxvf latest-64bit-<ARCH>-<OpenSSL>
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` nach `/usr/lib64/php/modules`.

```
sudo mv amazon-elasticache-cluster-client.so /usr/lib64/php/modules/
```

7. Hinzufügen von `extension=amazon-elasticache-cluster-client.so` in Datei `/etc/php.ini`
8. Wenn Sie den ElastiCache-Cluster-Client mit PHP 7.4 oder höher heruntergeladen haben, installieren Sie OpenSSL 1.1.x oder höher. Installationsanweisungen für OpenSSL 1.1.1:

```
sudo yum -y update
sudo yum install -y make gcc perl-core pcre-devel wget zlib-devel
wget https://www.openssl.org/source/openssl-1.1.1c.tar.gz
tar xvf openssl-1.1.1c.tar.gz
cd openssl-1.1.1c
./config
make
sudo make install
sudo ln -s /usr/local/lib64/libssl.so.1.1 /usr/lib64/libssl.so.1.1
```

Installieren Sie PHP 7.x bis 8.x auf einem Amazon Linux 201609 AMI wie folgt:

Note

Ersetzen Sie *php7.x* wenn nötig mit der Version, die Sie verwenden.

1. Starten Sie eine neue Instance aus dem AMI. Weitere Informationen finden Sie unter [Schritt 1: Starten einer Instance](#) im Benutzerhandbuch zu Amazon EC2.
2. Führen Sie den Befehl aus:

```
sudo yum install gcc-c++
```

3. Installieren von PHP

```
sudo yum install php7.x
```

4. Laden Sie den Amazon ElastiCache-Cluster-Client herunter.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.x/  
latest-64bit
```

5. Extrahieren Sie latest-64bit.

```
tar -zxvf latest-64bit
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei amazon-elasticache-cluster-client.so nach /usr/lib64/php/7.x/modules/.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.x/modules/
```

7. Erstellen Sie die Datei 50-memcached.ini.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /  
etc/php-7.x.d/50-memcached.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Installieren Sie PHP 7.x bis 8.x auf einem SUSE-Linux-15-AMI wie folgt:

Note

Ersetzen Sie *php7.x* wenn nötig mit der Version, die Sie verwenden.

1. Starten Sie eine neue Instance aus dem AMI.
2. Führen Sie den Befehl aus:

```
sudo zypper refresh
```



```
sudo zypper update -y
sudo zypper install gcc
```

3. Installieren von PHP

```
sudo yum install php7.x
```

oder

```
sudo zypper addrepo //download.opensuse.org/repositories/devel:/languages:/php/
openSUSE_Leap_15.3/ php
```

4. Laden Sie den Amazon-ElastiCache-Cluster-Client herunter und ersetzen Sie <ARCH> durch die gewünschte Architektur (X86 oder arm). SUSE 15 wird mit OpenSSL1.1 geliefert. Wählen Sie daher für PHP >= 7.4 die Client-Binärdatei mit OpenSSL1. Wenn Sie PHP < 7.4 verwenden, entfernen Sie das OpenSSL-Suffix.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.x/
latest-64bit-<ARCH>-openssl1.1
```

5. Extrahieren Sie latest-64bit.

```
tar -zxvf latest-64bit-<ARCH>-openssl1.1
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei amazon-elasticache-cluster-client.so nach /usr/lib64/php7/extensions/.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php7/extensions/
```


7. Fügen Sie die Zeile extension=amazon-elasticache-cluster-client.so in die Datei /etc/php7/cli/php.ini ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/
php7/cli/php.ini
```

8. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Installieren Sie PHP 7.x bis 8.x auf einem Ubuntu-22.04-AMI wie folgt:

 Note

Ersetzen Sie *php7.x* wenn nötig mit der Version, die Sie verwenden.

1. Starten Sie eine neue Instance aus dem AMI.
2. Führen Sie den Befehl aus:

```
sudo apt-get update
sudo apt-get install gcc g++ make zlib1g zlib1g-dev
```

3. Installieren von PHP

- a. Installationsanweisungen für PHP 8.1:

```
sudo apt install php8.1-cli php8.1-dev
```

- b. Installationsanweisungen für PHP 7.4:

```
sudo apt -y install software-properties-common
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
sudo apt -y install php7.4
```

4. Laden Sie den Amazon-ElastiCache-Cluster-Client herunter und ersetzen Sie <ARCH> durch die gewünschte Architektur (X86 oder arm). Ubuntu 22.04 wird mit OpenSSL3 geliefert. Wählen Sie daher Sie für PHP >= 7.4 die Client-Binärdatei mit OpenSSL3. Wenn Sie PHP < 7.4 verwenden, entfernen Sie das OpenSSL-Suffix.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.x/
latest-64bit-<ARCH>-openssl3
```

5. Extrahieren Sie die neueste 64-Bit-Version.

```
tar -zxvf latest-64bit-<ARCH>-openssl3
```

6. Kopieren Sie mit Root-Berechtigungen die extrahierte Artefaktdatei `amazon-elasticache-cluster-client.so` in das PHP-Erweiterungsverzeichnis `/usr/lib/php/20190902`. Falls dieses Erweiterungsverzeichnis nicht existiert, können Sie es finden, indem Sie Folgendes ausführen: `php -i | grep extension_dir`
7. Fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei `/etc/php/7.x/cli/php.ini` ein.

Installieren von PHP 5.x für neue Benutzer

Themen

- [So installieren Sie PHP 5 auf einem Amazon Linux 2014.03-AMI \(64-Bit und 32-Bit\)](#)
- [So installieren Sie PHP 5 auf einem Red Hat Enterprise Linux 7.0-AMI \(64-Bit und 32-Bit\)](#)
- [So installieren Sie PHP 5 auf einem Ubuntu-Server-14.04-LTS-AMI \(64-Bit und 32-Bit\)](#)
- [So installieren Sie PHP 5 für SUSE-Linux-Enterprise-Server-11-AMI \(64-Bit oder 32-Bit\)](#)
- [Andere Linux-Distributionen](#)

So installieren Sie PHP 5 auf einem Amazon Linux 2014.03-AMI (64-Bit und 32-Bit)

1. Starten Sie eine Amazon Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHP-Abhängigkeiten installieren:

```
$ sudo yum install gcc-c++ php php-pear
```

3. Laden Sie das korrekte `php-memcached`-Paket für Ihre Amazon-EC2-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren `php-memcached`. Die URI ist der Download-Pfad für das Installationspaket:

```
$ sudo pecl install <package download path>
```

Hier ist ein Beispielinstructionsbefehl für PHP 5.4, 64-Bit-Linux. Ersetzen Sie in diesem Beispiel `X.Y.Z` durch die tatsächliche Versionsnummer:

```
$ sudo pecl install /home/AmazonElastiCacheClusterClient-X.Y.Z-PHP54-64bit.tgz
```

Note

Verwenden Sie immer die aktuelle Version des Installationsartefakts.

- Erstellen Sie mit Root-/Sudo-Berechtigung eine neue Datei namens `memcached.ini` im Verzeichnis `/etc/php.d` und fügen Sie die Zeile "extension=amazon-elasticache-cluster-client.so" in die Datei ein:

```
$ echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php.d/memcached.ini
```

- Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 5 auf einem Red Hat Enterprise Linux 7.0-AMI (64-Bit und 32-Bit)

- Starten Sie eine Red Hat Enterprise Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
- PHP-Abhängigkeiten installieren:

```
sudo yum install gcc-c++ php php-pear
```

- Laden Sie das korrekte `php-memcached`-Paket für Ihre Amazon-EC2-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
- Installieren `php-memcached`. Die URI ist der Download-Pfad für das Installationspaket:

```
sudo pecl install <package download path>
```

- Erstellen Sie mit Root-/Sudo-Berechtigung eine neue Datei namens `memcached.ini` im Verzeichnis `/etc/php.d` und fügen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` in die Datei ein.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php.d/memcached.ini
```

- Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 5 auf einem Ubuntu-Server-14.04-LTS-AMI (64-Bit und 32-Bit)

1. Starten Sie eine Ubuntu Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHP-Abhängigkeiten installieren:

```
sudo apt-get update  
sudo apt-get install gcc g++ php5 php-pear
```

3. Laden Sie das korrekte php-memcached-Paket für Ihre Amazon-EC2-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren php-memcached. Die URI ist der Download-Pfad für das Installationspaket.

```
$ sudo pecl install <package download path>
```

Note

In diesem Installationsschritt wird das Build-Artefakt `amazon-elasticache-cluster-client.so` im Verzeichnis `/usr/lib/php5/20121212*` installiert. Überprüfen Sie den absoluten Pfad des Build-Artefakts, da er im nächsten Schritt benötigt wird.

Wenn der vorherige Befehl fehlschlägt, müssen Sie das PHP-Client-Artefakt `amazon-elasticache-cluster-client.so` manuell aus der heruntergeladenen Datei `*.tgz` extrahieren und in das Verzeichnis `/usr/lib/php5/20121212*` kopieren.

```
$ tar -xvf <package download path>  
cp amazon-elasticache-cluster-client.so /usr/lib/php5/20121212/
```

5. Erstellen Sie mit Root-/Sudo-Berechtigung eine neue Datei namens `memcached.ini` im Verzeichnis `/etc/php5/cli/conf.d` und fügen Sie die Zeile `"extension=<absolute path to amazon-elasticache-cluster-client.so>"` in die Datei ein.

```
$ echo "extension=<absolute path to amazon-elasticache-cluster-client.so>" | sudo tee --append /etc/php5/cli/conf.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

So installieren Sie PHP 5 für SUSE-Linux-Enterprise-Server-11-AMI (64-Bit oder 32-Bit)

1. Starten Sie eine SUSE-Linux-Instance (64-Bit oder 32-Bit) und melden Sie sich an.
2. PHP-Abhängigkeiten installieren:

```
$ sudo zypper install gcc php53-devel
```

3. Laden Sie das korrekte php-memcached-Paket für Ihre Amazon-EC2-Instance und PHP-Version herunter. Weitere Informationen finden Sie unter [Herunterladen des Installationspakets](#).
4. Installieren php-memcached. Die URI ist der Download-Pfad für das Installationspaket.

```
$ sudo pecl install <package download path>
```

5. Erstellen Sie mit Root-/Sudo-Berechtigung eine neue Datei namens `memcached.ini` im Verzeichnis `/etc/php5/conf.d` und fügen Sie die Zeile **`extension=amazon-elasticache-cluster-client.so`** in die Datei ein.

```
$ echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php5/conf.d/memcached.ini
```

6. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Note

Sollte Schritt 5 für die vorherigen Plattformen nicht funktionieren, überprüfen Sie den Installationspfad für `amazon-elasticache-cluster-client.so`. Geben Sie auch den

vollständigen Pfad der Binary in der Erweiterung an. Bei der verwendeten PHP-Version muss es sich um eine unterstützte Version handeln. Die Versionen 5.3 bis 5.5 werden unterstützt.

Andere Linux-Distributionen

In einigen Systemen, darunter CentOS7 und Red Hat Enterprise Linux (RHEL) 7.1 wurde `libsasl2.so.2` durch `libsasl2.so.3` ersetzt. In anderen Systemen kann die Datei `libsasl2.so.2` beim Laden des ElastiCache-Cluster-Clients nicht gefunden und geladen werden. Um dieses Problem zu beheben, erstellen Sie einen symbolischen Link auf `libsasl2.so.3`, damit der Client beim Laden von `libsasl2.so.2` auf `libsasl2.so.3` weitergeleitet wird. Mit dem folgenden Code wird dieser symbolische Link erstellt.

```
cd /usr/lib64
$ sudo ln libsasl2.so.3 libsasl2.so.2
```

Entfernen des PHP-Cluster-Clients

Themen

- [Entfernen einer früheren Version von PHP 7 oder höher](#)
- [Entfernen einer früheren Version von PHP 5](#)

Entfernen einer früheren Version von PHP 7 oder höher

Entfernen Sie eine frühere Version von PHP 7 oder höher wie folgt:

1. Entfernen Sie die Datei `amazon-elasticache-cluster-client.so` aus dem entsprechenden PHP-Bibliotheksverzeichnis, wie zuvor in den Installationsanleitungen erläutert. Weitere Informationen finden Sie in Installationsabschnitt unter [Für Benutzer, die bereits die Erweiterung php-memcached installiert haben](#).
2. Entfernen Sie die Zeile `extension=amazon-elasticache-cluster-client.so` aus der Datei `php.ini`.
3. Starten Sie Ihren Apache-Server (neu).

```
sudo /etc/init.d/httpd start
```

Entfernen einer früheren Version von PHP 5

So entfernen Sie eine frühere Version von PHP 5

1. Entfernen Sie die Erweiterung php-memcached:

```
sudo pecl uninstall __uri/AmazonElastiCacheClusterClient
```

2. Entfernen Sie die Datei `memcached.ini` aus dem entsprechenden Verzeichnis, wie zuvor in den Installationsanleitungen erläutert.

Kompilieren des Quellcodes für den ElastiCache-Cluster-Client für PHP

Dieser Abschnitt beschreibt, wie Sie den Quellcode für den ElastiCache-Cluster-Client-für-PHP erhalten und kompilieren.

Sie müssen zwei Pakete von GitHub per Pull ausführen und kompilieren: [aws-elasticache-cluster-client-libmemcached](#) und [aws-elasticache-cluster-client-memcached-for-php](#).

Themen

- [Kompilieren der libmemcached-Bibliothek](#)
- [Kompilieren des ElastiCache-Memcached-Auto-Discovery-Clients für PHP](#)

Kompilieren der libmemcached-Bibliothek

Voraussetzungen für Bibliotheken

- OpenSSL 1.1.0 oder höher (es sei denn, die TLS-Unterstützung ist von `./configure --disable-tls` deaktiviert).
- SASL (libsasl2), es sei denn, die SASL-Unterstützung ist von `./configure --disable-sasl` deaktiviert).

So kompilieren Sie die `aws-elasticache-cluster-client-libmemcached`-Bibliothek

1. Starten Sie eine Amazon-EC2-Instance.
2. Installieren Sie die Abhängigkeiten der Bibliothek.
 - Auf dem Amazon Linux 201509 AMI/dem Amazon Linux 2 AMI

```
sudo yum -y update
sudo yum install gcc gcc-c++ autoconf libevent-devel make perl-core pcre-devel
wget zlib-devel
// Install OpenSSL 1.1.1
wget https://www.openssl.org/source/openssl-1.1.1c.tar.gz
tar xvf openssl-1.1.1c.tar.gz
cd openssl-1.1.1c
./config
make
sudo make install
sudo ln -s /usr/local/lib64/libssl.so.1.1 /usr/lib64/libssl.so.1.1
```

- Auf Ubuntu 14.04 AMI (nicht erforderlich für Ubuntu-Versionen, die mit OpenSSL \geq 1.1 geliefert werden)

```
sudo apt-get update

sudo apt-get install libevent-dev gcc g++ make autoconf libsasl2-dev

// Install OpenSSL 1.1.1
wget https://www.openssl.org/source/openssl-1.1.1c.tar.gz
tar xvf openssl-1.1.1c.tar.gz
cd openssl-1.1.1c
./config
make
sudo make install
sudo ln -s /usr/local/lib/libssl.so.1.1 /usr/lib/x86_64-linux-gnu/libssl.so.1.1
```

3. Führen Sie einen Pull des Repositorys aus und kompilieren Sie den Code.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-
libmemcached.git
cd aws-elasticache-cluster-client-libmemcached
touch configure.ac aclocal.m4 configure Makefile.am Makefile.in
mkdir BUILD
cd BUILD
../configure --prefix=<libmemcached-install-directory> --with-pic --disable-sasl
```

Wenn bei der Ausführung von `../configure libssl` (OpenSSL-Bibliothek) nicht gefunden werden kann, kann es notwendig sein, die Umgebungsvariable `PKG_CONFIG_PATH` anzupassen:

```
PKG_CONFIG_PATH=/path/to/ssl/lib/pkgconfig ../configure --prefix=<libmemcached-
install-directory> --with-pic --disable-sasl
```

Wenn Sie TLS nicht verwenden, können Sie es auch deaktivieren, indem Sie Folgendes ausführen:

```
make
sudo make install
../configure --prefix=<libmemcached-install-directory> --with-pic --disable-sasl --
disable-tls
```

Kompilieren des ElastiCache-Memcached-Auto-Discovery-Clients für PHP

In den folgenden Abschnitten wird beschrieben, wie Sie den ElastiCache-Memcached-Auto-Discovery-Client kompilieren

Themen

- [Kompilieren des ElastiCache-Memcached-Clients für PHP 7 oder höher](#)
- [Kompilieren des ElastiCache-Memcached-Clients für PHP 5](#)

Kompilieren des ElastiCache-Memcached-Clients für PHP 7 oder höher

Ersetzen Sie PHP-7.x mit der Version, die Sie verwenden.

Installieren von PHP:

```
sudo yum install -y amazon-linux-extras
sudo amazon-linux-extras enable php7.x
```

Führen Sie die folgenden Befehle im Code-Verzeichnis aus.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git
cd aws-elasticache-cluster-client-memcached-for-php
phpize
mkdir BUILD
CD BUILD
../configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-memcached-sasl
```

Wenn die Ausführung von `../configure libssl` (OpenSSL-Bibliothek) nicht finden kann, könnte es notwendig sein, die `PKG_CONFIG_PATH` -Umgebungsvariable im `.PC`-Dateiverzeichnis von OpenSSL zu ändern:

```
PKG_CONFIG_PATH=/path/to/ssl/lib/pkgconfig ../configure --with-libmemcached-dir=<path to libmemcached build directory> --disable-memcached-sasl
```

Wenn Sie TLS nicht verwenden, können Sie es auch deaktivieren, indem Sie Folgendes ausführen:

```
make
```

```
make install
../configure --with-libmemcached-dir=<path to libmemcached build directory> --disable-
memcached-sasl --disable-memcached-tls
```

Note

Sie können die libmemcached-Bibliothek mit der PHP-Binärdatei statisch verknüpfen, sodass sie über verschiedene Plattformen portiert werden kann. Führen Sie dazu den folgenden Befehl vor make aus:

```
sed -i "s#-lmemcached#<libmemcached-install-directory>/lib/libmemcached.a -
lcrypt -lpthread -lm -lstdc++ -lsasl2#" Makefile
```

Kompilieren des ElastiCache-Memcached-Clients für PHP 5

Kompilieren Sie den `aws-elasticache-cluster-client-memcached-for-php`, indem Sie die folgenden Befehle im Ordner `aws-elasticache-cluster-client-memcached-for-php/` ausführen.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-
php/tree/php.git
cd aws-elasticache-cluster-client-memcached-for-php
sudo yum install zlib-devel
phpize
./configure --with-libmemcached-dir=<libmemcached-install-directory>
make
make install
```

Konfiguration von ElastiCache Clients

Ein ElastiCache Cluster ist protokollkonform mit Memcached. Der Code, die Anwendungen und die meisten Tools, die Sie heute in Ihrer vorhandenen Memcached-Umgebung verwenden, werden daher nahtlos mit dem Service zusammenarbeiten.

In diesem Abschnitt werden spezifische Überlegungen zum Herstellen einer Verbindung zu Cache-Knoten in beschrieben. ElastiCache

Themen

- [Finden von Knotenendpunkten und Portnummern](#)
- [Verbinden für die Verwendung der automatischen Erkennung](#)
- [DNS-Namen und zugrunde liegende IP](#)

Finden von Knotenendpunkten und Portnummern

Um eine Verbindung zu einem Cache-Knoten herzustellen, muss Ihre Anwendung den Endpunkt und die Portnummer für diesen Knoten kennen.

Finden von Knotenendpunkten und Portnummern (Konsole)

So bestimmen Sie Knotenendpunkte und Portnummern

1. Melden Sie sich bei der [Amazon-ElastiCache-Managementkonsole](#) an und wählen Sie die Engine, die auf Ihrem Cluster läuft.

Es wird eine Liste aller Cluster der ausgewählten Engine angezeigt.

2. Fahren Sie mit der Anleitung für die ausgewählte Engine und Konfiguration fort.
3. Wählen Sie den Namen des gewünschten Clusters aus.
4. Entnehmen Sie die benötigten Informationen für den gewünschten Knoten aus den Spalten Port und Endpoint.

Finden von Cache-Knotenendpunkten und Portnummern (AWS CLI)

Verwenden Sie den Befehl `describe-cache-clusters` mit dem Parameter `--show-cache-node-info`, um die Endpunkte und Portnummern von Cache-Knoten zu bestimmen.

```
aws elasticache describe-cache-clusters --show-cache-node-info
```

Sie finden die vollständig qualifizierten DNS-Namen und Portnummern im Abschnitt Endpunkt in der Ausgabe.

Finden von Cache-Knotenendpunkten und Portnummern (ElastiCache API)

Verwenden Sie die Aktion `DescribeCacheClusters` mit dem Parameter `ShowCacheNodeInfo=true`, um die Endpunkte und Portnummern von Cache-Knoten zu bestimmen.

Example

```
https://elasticache.us-west-2.amazonaws.com /
  ?Action=DescribeCacheClusters
  &ShowCacheNodeInfo=true
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20140421T220302Z
  &Version=2014-09-30
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
  &X-Amz-Credential=<credential>
  &X-Amz-Date=20140421T220302Z
  &X-Amz-Expires=20140421T220302Z
  &X-Amz-Signature=<signature>
  &X-Amz-SignedHeaders=Host
```

Verbinden für die Verwendung der automatischen Erkennung

Wenn Sie in Ihren Anwendungen Auto Discovery verwenden, müssen Sie anstelle der einzelnen Endpunkte für jeden Cache-Knoten den Konfigurationsendpunkt für das Cluster kennen. Weitere Informationen finden Sie unter [Automatische Erkennung von Knoten in Ihrem Cluster](#).

Note

Auto Discovery ist derzeit nur für Cache-Cluster verfügbar, die in Memcached ausgeführt werden.

DNS-Namen und zugrunde liegende IP

Clients verfügen über eine Serverliste mit den Adressen und Ports der Server, auf denen Cache-Daten abgelegt werden. Bei Verwendung von ElastiCache gibt die API `DescribeCacheClusters` (oder

das Befehlszeilenprogramm `describe-cache-clusters`) einen voll qualifizierten DNS-Eintrag und eine Portnummer zurück, die für die Serverliste verwendet werden können.

Important

Es ist wichtig, dass Client-Anwendungen so konfiguriert werden, dass DNS-Namen von Cache-Knoten regelmäßig aufgelöst werden, wenn sie versuchen, eine Verbindung zu einem Cache-Knotenendpunkt herzustellen.

VPC-Installationen

ElastiCache stellt sicher, dass sowohl der DNS-Name als auch die IP-Adresse des Cache-Knotens gleich bleiben, wenn Cache-Knoten im Falle eines Ausfalls wiederhergestellt werden.

Nicht-VPC-Installationen

ElastiCache stellt sicher, dass der DNS-Name eines Cache-Knotens unverändert bleibt, wenn Cache-Knoten im Falle eines Ausfalls wiederhergestellt werden; die zugrunde liegende IP-Adresse des Cache-Knotens kann sich jedoch ändern.

Persistente Cache-Knotenverbindungen werden von den meisten Client-Bibliotheken standardmäßig unterstützt. Wir empfehlen, bei der Verwendung von ElastiCache persistente Cache-Knotenverbindungen zu verwenden. Clientseitige Zwischenspeicherung von DNS-Informationen kann an mehreren Orten erfolgen, darunter Client-Bibliotheken, der Sprache der Laufzeitumgebung oder dem Betriebssystem des Clients. Prüfen Sie die Konfiguration Ihrer Anwendung auf allen Ebenen, um sicherzustellen, dass IP-Adressen für Ihre Cache-Knoten regelmäßig aufgelöst werden.

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 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:

Möchten Sie einen ElastiCache serverlosen oder instanzbasierten Dienst verwenden?

Wenn Sie Serverless-Cache verwenden möchten, stellen Sie einfach sicher, dass Sie Ihre VPC, Subnetze und Sicherheitsgruppen ordnungsgemäß konfiguriert haben. Weitere Details finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#). Wenn Sie einen instanzbasierten Dienst verwenden möchten ElastiCache, lesen Sie weiter.

- Welchen Knoten-Instance-Typ benötigen Sie?

Eine Anleitung bei der Auswahl eines Knoten-Instance-Typs finden Sie unter [Auswahl der Memcached-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 so konzipiert, dass auf Amazon EC2 von innen AWS zugegriffen werden kann. 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 [Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC](#).

- Müssen Sie irgendwelche Parameterwerte anpassen?

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

- 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 Memcached-Cluster](#)
- [Voraussetzungen für Skalierung](#)
- [Zugriffsvoraussetzungen](#)
- [Anforderungen für Region und Availability Zone](#)

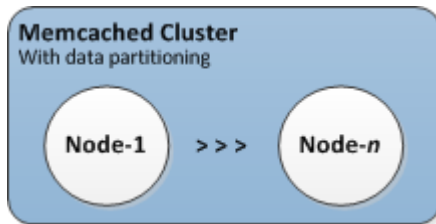
Anforderungen an Speicher und Prozessor

Der Grundbaustein von Amazon ElastiCache ist der Node. Knoten werden zur Bildung von Clustern einzeln oder in Gruppierungen konfiguriert. Berücksichtigen Sie bei der Bestimmung des für den Cluster zu verwendenden Knotentyps die Knotenkonfiguration des Clusters und die zu speichernde Datenmenge.

Die Memcached-Engine arbeitet mit mehreren Threads. Daher wirkt sich die Anzahl der Kerne eines Knotens auf die für den Cluster verfügbare Rechenleistung aus.

Konfiguration von Memcached-Cluster

ElastiCache (Memcached-) Cluster bestehen aus 1 bis 60 Knoten. Die Daten in einem Memcached-Cluster werden über Knoten im Cluster hinweg partitioniert. Ihre Anwendung stellt über eine als Endpunkt bezeichnete Netzwerkadresse eine Verbindung mit einem Memcached-Cluster her. Jeder Knoten in einem Memcached-Cluster verfügt über seinen eigenen Endpunkt. Dieser wird von der Anwendung zum Lesen aus oder Schreiben in einem bestimmten Knoten genutzt. Zusätzlich zu den Knotenendpunkten hat der Memcached-Cluster selbst einen Endpunkt, der als Konfigurationsendpunkt bezeichnet wird. Ihre Anwendung kann diesen Endpunkt verwenden, um aus dem Cluster zu lesen oder in den Cluster zu schreiben, wobei die Bestimmung des Knotens, aus dem gelesen oder in den geschrieben werden soll, [Automatische Erkennung von Knoten in Ihrem Cluster](#) überlassen bleibt.



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 Memcached-Cluster hochskalieren, ist der neue Cluster zunächst leer.

Amazon ElastiCache for Memcached-Cluster können horizontal oder horizontal skaliert werden. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten aus dem Cluster nach oben bzw. nach unten skaliert. Wenn Automatic Discovery aktiviert ist und die Anwendung mit dem Konfigurationsendpunkt des Clusters verbunden ist, sind zum Hinzufügen oder Entfernen von Knoten keine Änderungen an Ihrer Anwendung erforderlich.

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

Zugriffsvoraussetzungen

Der Zugriff auf ElastiCache Amazon-Cluster erfolgt standardmäßig über Amazon EC2-Instances. Der Netzwerkzugriff auf einen ElastiCache Cluster ist auf das Konto beschränkt, mit dem der Cluster erstellt wurde. 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 mit der Instance verknüpft ist, Zugriff auf Ihre ElastiCache Sicherheitsgruppe gewähren. Detaillierte Anweisungen finden Sie unter [Zugriff auf Ihren Cluster oder die](#) 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:

- [Regionen und Availability Zones](#)
- [Lokale Zonen](#)
- [Minimieren von Ausfällen](#)

Auswahl der Knotengröße

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

Auswahl der Memcached-Knotengröße

Memcached-Cluster enthalten gegebenenfalls mehrere Knoten mit über die Knoten partitionierten Cluster-Daten. Aus diesem Grund sind die Speicheranforderungen des Clusters und die eines Knotens zwar ähnlich, jedoch nicht identisch. Sie können Ihre erforderliche Cluster-Speicherkapazität mit wenigen großen Knoten oder mit vielen kleineren Knoten abdecken. Mit wechselnden Anforderungen können Sie Knoten zum Cluster hinzufügen oder aus diesem entfernen und zahlen so nur für die Knoten, die Sie tatsächlich brauchen.

Die Gesamtspeicherkapazität des Clusters wird durch Multiplizieren der Anzahl von Knoten im Cluster mit der RAM-Kapazität der einzelnen Knoten (nach Abzug der für die Systemverwaltung erforderlichen Kapazität) berechnet. Die Kapazität jedes Knotens basiert auf dem Knotentyp.

```
cluster_capacity = number_of_nodes * (node_capacity - system_overhead)
```

Die Anzahl von Knoten im Cluster ist ein Schlüsselfaktor für die Verfügbarkeit Ihres Clusters, der Memcached ausführt. Der Ausfall eines einzelnen Knotens kann Auswirkungen auf die Verfügbarkeit Ihrer Anwendung und die Belastung Ihrer Backend-Datenbank haben. In einem solchen Fall stellt es einen ElastiCache Ersatz für einen ausgefallenen Knoten bereit und dieser wird erneut aufgefüllt. Um diese Auswirkungen auf die Verfügbarkeit zu verringern, sollten Sie die Speicher- und Rechenkapazität auf mehrere Knoten mit geringerer Kapazität verteilen, anstatt weniger Knoten mit hoher Kapazität zu verwenden.

In einem Szenario, in dem Sie 35 GB Cache-Speicher haben möchten, können Sie eine der folgenden Konfigurationen einrichten:

- 11 `cache.t2.medium`-Knoten mit 3,22 GB Speicher und jeweils 2 Threads = 35,42 GB und 22 Threads.
- 6 `cache.m4.large`-Knoten mit 6,42 GB Speicher und jeweils 2 Threads = 38,52 GB und 12 Threads.
- 3 `cache.r4.large`-Knoten mit 12,3 GB Speicher und jeweils 2 Threads = 36,90 GB und 6 Threads.
- 3 `cache.m4.xlarge`-Knoten mit 14,28 GB Speicher und jeweils 4 Threads = 42,84 GB und 12 Threads.

Vergleichen der Knotenoptionen

Knotentyp	Speicher (in GiB)	Kerne	Kosten pro Stunde*	Erforderliche Knoten	Gesamtspeicher (in GiB)	Kerne gesamt	Monatlicher Preis
<code>cache.t2.medium</code>	3,22	2	0,068 USD	11	35,42	22	538,56 USD
<code>cache.m4.large</code>	6,42	2	0,156 USD	6	38,52	12	673,92 USD
<code>cache.m4.xlarge</code>	14,28	4	0,311 USD	3	42,84	12	671,76 USD
<code>cache.m5.xlarge</code>	12,93	4	0,311 USD	3	38,81	12	671,76 USD
<code>cache.m6g.large</code>	6,85	2	\$0,147	6	41,1	12	\$635
<code>cache.r4.large</code>	12,3	2	0,228 USD	3	36,9	6	492,48 USD
<code>cache.r5.large</code>	13,07	2	0,216 USD	3	39,22	6	466,56 USD
<code>cache.r6g.large</code>	13,07	2	\$0,205	3	42,12	6	\$442

Knotentyp	Speicher (in GiB)	Kerne	Kosten pro Stunde*	Erforderliche Knoten	Gesamtspeicher (in GiB)	Kerne gesamt	Monatlicher Preis
-----------	----------------------	-------	--------------------------	-------------------------	-------------------------------	-----------------	----------------------

* Preis pro Stunde und Knoten ab 08. Oktober 2020.

Kosten pro Monat bei 100 %-iger Auslastung für 30 Tage (720 Stunden).

Diese Optionen bieten jeweils ähnliche Speicherkapazität bei unterschiedlicher Rechenkapazität und Kosten. Um die Kosten Ihrer spezifischen Optionen zu vergleichen, sehen Sie sich die [ElastiCache Amazon-Preise](#) an.

Für Cluster, die Memcached ausführen, wird ein Teil des verfügbaren Speichers auf jedem Knoten für den Verbindungs-Overhead verwendet. Weitere Informationen finden Sie unter [Overhead von Memcached-Verbindungen](#).

Bei der Verwendung mehrerer Knotenpunkte müssen die Schlüssel auf diese verteilt werden. Jeder Knoten besitzt einen eigenen Endpunkt. Für eine einfache Endpunktverwaltung können Sie ElastiCache die Auto Discovery-Funktion verwenden, mit der Client-Programme automatisch alle Knoten in einem Cluster identifizieren können. Weitere Informationen finden Sie unter [Automatische Erkennung von Knoten in Ihrem Cluster](#).

In manchen Fällen sind Sie vielleicht unsicher, wie viel Kapazität Sie benötigen. Wenn das so ist, empfehlen wir, zu Testzwecken mit einem `cache.m5.large`-Knoten zu beginnen. Überwachen Sie dann die Speicherauslastung, die CPU-Auslastung und die Cache-Trefferquote anhand der auf Amazon veröffentlichten ElastiCache Metriken CloudWatch. Weitere Informationen zu CloudWatch Metriken für ElastiCache finden Sie unter [Überwachung der Nutzung mithilfe von CloudWatch-Metriken](#). Für die Produktion und größere Workloads bieten R5-Knoten die beste Leistung und einen optimalen RAM-Einstandswert.

Wenn Ihr Cluster nicht die gewünschte Trefferquote aufweist, können Sie einfach weitere Knoten hinzufügen, um den gesamten verfügbaren Speicher in Ihrem Cluster zu erhöhen.

Wenn Ihr Cluster durch die CPU begrenzt ist, aber eine ausreichende Trefferquote aufweist, richten Sie einen neuen Cluster mit einem Knotentyp ein, der mehr Rechenleistung bietet.

Erstellen eines Clusters

Die folgenden Beispiele zeigen, wie Sie mithilfe der ElastiCache API, der AWS Management Console, der AWS CLI und eines Clusters erstellen.

Erstellen eines Memcached-Clusters (Konsole)

Wenn Sie die Memcached-Engine verwenden, unterstützt Amazon die horizontale Partitionierung Ihrer Daten über mehrere Knoten. Da Memcached Auto Discovery ermöglicht, brauchen Sie die Endpunkte eines jeden Knotens nicht zu verfolgen. Memcached verfolgt den Endpunkt jedes Knotens und aktualisiert die Endpunktliste beim Hinzufügen oder Entfernen von Knoten. Ihre Anwendung braucht nur mit dem Cluster zu kommunizieren, der als Konfigurationsendpunkt fungiert. Weitere Informationen zu Auto Discovery finden Sie unter [Automatische Erkennung von Knoten in Ihrem Cluster](#).

Befolgen Sie zum Erstellen eines Memcached-Clusters die Schritte unter [Schritt 1: Erstellen eines Caches](#).

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 [Zugriff auf Ihren Cluster oder die](#) und [Manuelles Herstellen von Verbindungen mit Cache-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, um einen Cluster mit dem AWS CLI 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 Memcached-Cache-Clusters (AWS CLI)

Der folgende CLI-Code erstellt einen Memcached-Cache-Cluster mit 3 Knoten.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  
--cache-node-type cache.r4.large \  
--engine memcached \  
--engine-version 1.4.24 \  
--cache-parameter-group default.memcached1.4 \  
--num-cache-nodes 3
```

Für Windows:

```
aws elasticache create-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--cache-node-type cache.r4.large ^  
--engine memcached ^  
--engine-version 1.4.24 ^  
--cache-parameter-group default.memcached1.4 ^  
--num-cache-nodes 3
```

Einen Cluster (ElastiCache API) erstellen

Verwenden Sie die `CreateCacheCluster` Aktion, um einen Cluster mithilfe der ElastiCache API zu erstellen.

Important

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

Einen Memcached-Cache-Cluster (API) ElastiCache erstellen

Der folgende Code erstellt einen Memcache-Cluster mit 3 Knoten (API). ElastiCache

Die Zeilenumbrüche dienen der besseren Lesbarkeit.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=CreateCacheCluster  
  &CacheClusterId=my-cluster  
  &CacheNodeType=cache.r4.large  
  &Engine=memcached  
  &NumCacheNodes=3  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150508T220302Z  
  &Version=2015-02-02  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Credential=<credential>  
  &X-Amz-Date=20150508T220302Z  
  &X-Amz-Expires=20150508T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Signature=<signature>
```


Anzeigen der Details eines Clusters

Sie können Detailinformationen zu einem oder mehreren Clustern mithilfe der ElastiCache Konsole oder der ElastiCache API anzeigen. AWS CLI

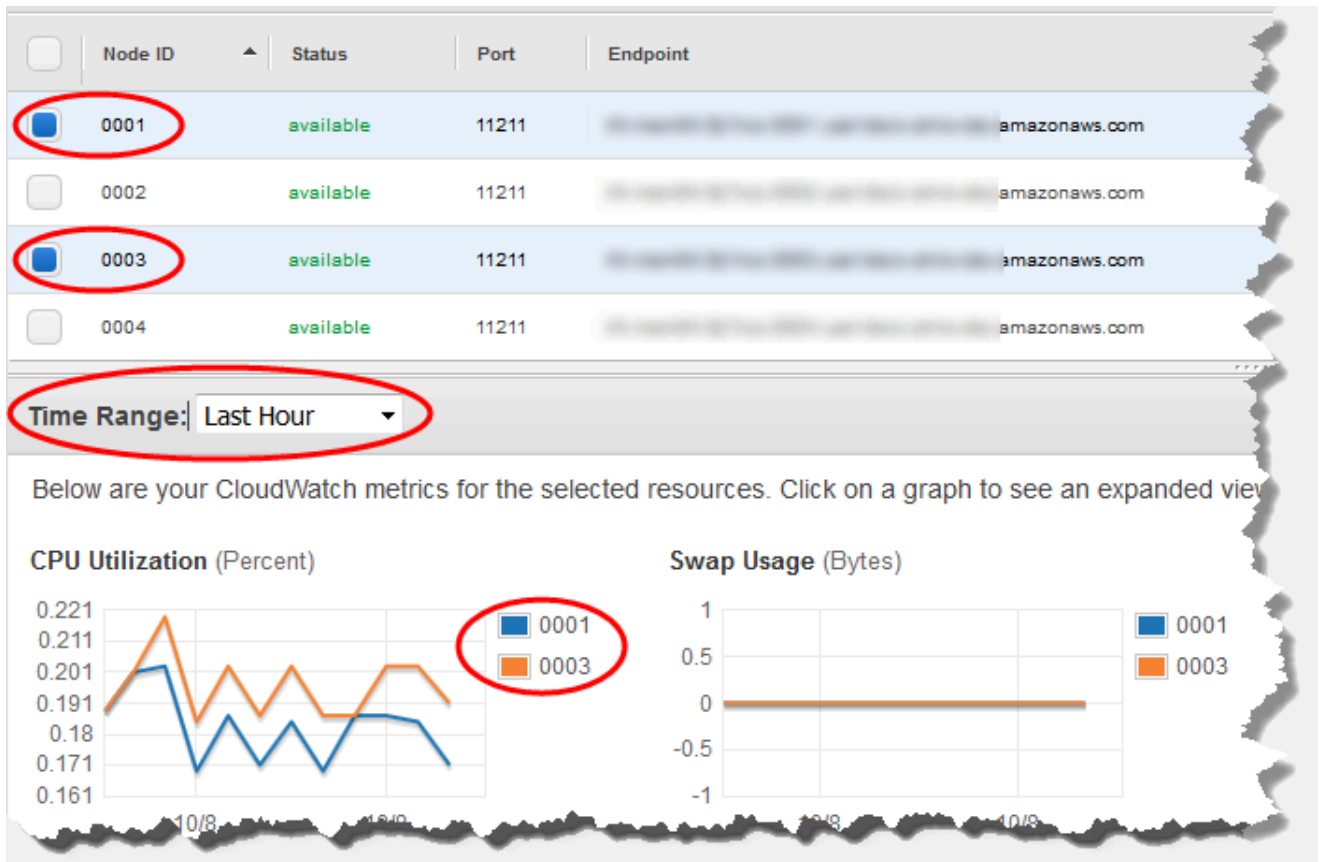
Anzeigen der Details eines Clusters (Konsole)

Sie können die Details eines Memcached-Clusters mithilfe der ElastiCache Konsole, des For-Clusters oder der AWS CLI ElastiCache ElastiCache API anzeigen.

Das folgende Verfahren beschreibt, wie Sie die Details eines Memcache-Clusters mithilfe der Konsole anzeigen. ElastiCache

So zeigen Sie die Details eines Memcached-Clusters an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie aus der Liste in der oberen rechten Ecke die AWS Region aus, für die Sie sich interessieren.
3. Wählen Sie im ElastiCache Konsolen-Dashboard Memcached aus. Damit werden alle Cluster aufgelistet, die eine Version von Memcached ausführen.
4. Um Details eines Clusters anzuzeigen, aktivieren Sie das Kontrollkästchen links neben den Namen des Clusters.
5. Anzeigen der Knoteninformationen:
 - a. Wählen Sie den Cluster-Namen aus.
 - b. Wählen Sie die Registerkarte Nodes aus.
 - c. Um die Metriken eines oder mehrerer Knoten anzuzeigen, aktivieren Sie das Kontrollkästchen links neben der Knoten-ID. Wählen Sie dann den Zeitraum für die Metriken aus der Liste Time range aus. Durch die Auswahl mehrerer Knoten werden Überlagerungsdiagramme generiert.



Metriken der letzten Stunde für zwei Memcached-Knoten

Anzeigen der Details eines Clusters (AWS CLI)

Sie können die Details für einen Cluster mit dem AWS CLI `describe-cache-clusters` Befehl anzeigen. Wenn der Parameter `--cache-cluster-id` weggelassen wird, werden Details für mehrere Cluster, bis zu `--max-items`, zurückgegeben. Wenn der Parameter `--cache-cluster-id` eingeschlossen wird, werden Details für den angegebenen Cluster zurückgegeben. Sie können die Anzahl der mit dem Parameter `--max-items` zurückgegebenen Datensätze begrenzen.

Der folgende Code listet die Details für `my-cluster` auf.

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

Der folgende Code listet die Details für bis zu 25 Cluster auf.

```
aws elasticache describe-cache-clusters --max-items 25
```

Example

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id my-cluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id my-cluster ^  
  --show-cache-node-info
```

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

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {  
          "CacheNodeId": "0001",  
          "Endpoint": {  
            "Port": 11211,  
            "Address": "my-cluster.7ef-  
example.0001.usw2.cache.amazonaws.com"  
          },  
          "CacheNodeStatus": "available",  
          "ParameterGroupStatus": "in-sync",  
          "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",  
          "CustomerAvailabilityZone": "us-west-2b"  
        },  
        {  
          "CacheNodeId": "0002",  
          "Endpoint": {  
            "Port": 11211,  
            "Address": "my-cluster.7ef-  
example.0002.usw2.cache.amazonaws.com"  
          },  
          "CacheNodeStatus": "available",  
          "ParameterGroupStatus": "in-sync",  
          "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",
```

```
        "CustomerAvailabilityZone": "us-west-2b"
    },
    {
        "CacheNodeId": "0003",
        "Endpoint": {
            "Port": 11211,
            "Address": "my-cluster.7ef-
example.0003.usw2.cache.amazonaws.com"
        },
        "CacheNodeStatus": "available",
        "ParameterGroupStatus": "in-sync",
        "CacheNodeCreateTime": "2016-09-21T16:28:28.973Z",
        "CustomerAvailabilityZone": "us-west-2b"
    }
],
"CacheParameterGroup": {
    "CacheNodeIdsToReboot": [],
    "CacheParameterGroupName": "default.memcached1.4",
    "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "my-cluster",
"PreferredAvailabilityZone": "us-west-2b",
"ConfigurationEndpoint": {
    "Port": 11211,
    "Address": "my-cluster.7ef-example.cfg.usw2.cache.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "available",
"NumCacheNodes": 3,
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
"SecurityGroups": [
    {
        "Status": "active",
        "SecurityGroupId": "sg-dbe93fa2"
    }
],
"CacheSubnetGroupName": "default",
"EngineVersion": "1.4.24",
"PendingModifiedValues": {},
"PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
"CacheNodeType": "cache.m3.medium"
```

```
    }  
  ]  
}
```

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

Details eines Clusters anzeigen (ElastiCache API)

Sie können die Details für einen Cluster mithilfe der ElastiCache DescribeCacheClusters API-Aktion anzeigen. Wenn der Parameter CacheClusterId eingeschlossen wird, werden Details für den angegebenen Cluster zurückgegeben. Wenn der Parameter CacheClusterId weggelassen wird, werden Details für bis zu MaxRecords (Standard 100) Cluster zurückgegeben. Der Wert für MaxRecords darf nicht kleiner als 20 oder größer als 100 sein.

Der folgende Code listet die Details für my-cluster auf.

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

Der folgende Code listet die Details für bis zu 25 Cluster auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&MaxRecords=25  
&Version=2015-02-02  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&X-Amz-Credential=<credential>
```

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

Einen ElastiCache Cluster ändern

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 der Spalte Changes Take Effect (Änderungen treten in Kraft) in den Tabellen für [Memcached-spezifische Parameter](#) und können Sie bestimmen, wann eine bestimmte Parameteränderung übernommen wird. Weitere Informationen zum Neustarten eines Clusters finden Sie unter [Neustart eines Clusters](#).

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:
 - Kompatibilität der Engine-Version
 - VPC Security group(s) (VPC-Sicherheitsgruppe(n))

- Parametergruppe
- Wartungsfenster
- 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 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.

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. 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 der Engine-Version und für die Änderung der Anzahl der 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. 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](#).

Neustart eines Clusters

Bei einigen Änderungen muss der Cluster neu gestartet werden, damit die Änderungen übernommen werden. Beispielsweise wird bei einigen Parametern eine Änderung am Parameterwert in einer Parametergruppe erst nach einem Neustart übernommen.

Wenn Sie einen Cluster neu starten, bereinigt der Cluster alle seine Daten und startet seine Engine neu. Während dieses Vorgangs können Sie nicht auf den Cluster zugreifen. Da der Cluster alle seine Daten bereinigt hat, starten Sie mit einem leeren Cluster, wenn der Cluster wieder verfügbar ist.

Sie können einen Cluster über die ElastiCache-Konsole, die AWS CLI oder die ElastiCache-API neu starten. Unabhängig davon, ob Sie die ElastiCache-Konsole, die AWS CLI oder die ElastiCache-API verwenden, können Sie nur den Neustart eines einzelnen Clusters initiieren. Um mehrere Cluster neu zu starten, müssen Sie den Prozess oder Vorgang iterieren.

Verwendung der AWS Management Console

Sie können einen Cluster mithilfe der ElastiCache-Konsole neu starten.

So starten Sie einen Cluster neu (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, die Sie interessiert.
3. Wählen Sie im Navigationsbereich die Engine, die auf dem Cluster ausgeführt wird, den Sie neu starten möchten.

Es wird eine Liste der Cluster der ausgewählten Engine angezeigt.

4. Wählen Sie den neu zu startenden Cluster aus, indem Sie das Kontrollkästchen links neben dem Clusternamen aktivieren.

Wählen Sie Aktionen und dann Neustart aus.

Wenn Sie mehr als einen Cluster auswählen, ist das Kontrollkästchen Reboot deaktiviert.

Wiederholen Sie zum Neustarten mehrerer Cluster die Schritte 2 bis 5 für jeden Cluster, den Sie neu starten möchten. Sie brauchen nicht zu warten, bis ein Cache-Cluster fertig neu gestartet wurde, bevor Sie einen anderen neu starten.

Um einen bestimmten Knoten neu zu starten, markieren Sie den Knoten und wählen Sie dann Neustart.

Verwendung der AWS CLI

Um einen Cluster neu zu starten (AWS CLI), verwenden Sie den `reboot-cache-cluster`-CLI-Vorgang.

Um bestimmte Knoten im Cluster neu zu starten, listen Sie mit `--cache-node-ids-to-reboot` die spezifischen Cluster auf, die neu gestartet werden sollen. Der folgende Befehl startet die Knoten 0001, 0002 und 0004 von `my-cluster` neu.

Für Linux, macOS oder Unix:

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

Für Windows:

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

Um alle Knoten im Cluster neu zu starten, listen Sie mit dem Parameter `--cache-node-ids-to-reboot` die Knoten-IDs aller Cluster auf. Weitere Informationen finden Sie unter [reboot-cache-cluster](#).

Verwenden der ElastiCache-API

Um einen Cluster über die ElastiCache-API neu zu starten, verwenden Sie die `RebootCacheCluster`-Aktion.

Um bestimmte Knoten im Cluster neu zu starten, listen Sie mit `CacheNodeIdsToReboot` die spezifischen Cluster auf, die neu gestartet werden sollen. Der folgende Befehl startet die Knoten 0001, 0002 und 0004 von `my-cluster` neu.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RebootCacheCluster  
&CacheClusterId=my-cluster
```

```
&CacheNodeIdsToReboot.member.1=0001
&CacheNodeIdsToReboot.member.2=0002
&CacheNodeIdsToReboot.member.3=0004
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Um alle Knoten im Cluster neu zu starten, listen Sie mit dem Parameter `CacheNodeIdsToReboot` die Knoten-IDs aller Cluster auf. Weitere Informationen finden Sie unter [RebootCacheCluster](#).

Knoten zu einem Cluster hinzufügen

Das Hinzufügen von Knoten zu einem Memcached-Cluster erhöht die Anzahl Ihrer Cluster-Partitionen. Wenn Sie die Anzahl der Partitionen in einem Cluster ändern, müssen einige Ihrer Schlüsselbereiche neu zugeordnet werden, damit sie auf den richtigen Knoten abgebildet werden. Die Neuordnung von Schlüsselräumen erhöht vorübergehend die Anzahl der Cache-Fehler auf dem Cluster. Weitere Informationen finden Sie unter [Konfigurieren Ihres ElastiCache-Clients für ein effizientes Load Balancing](#).

Sie können die ElastiCache Management Console, die AWS CLI oder ElastiCache API verwenden, um Ihrem Cluster Knoten hinzuzufügen.

Mit dem AWS Management Console

Themen

- [So fügen Sie Knoten zu einem Cluster hinzu \(Konsole\)](#)

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 <https://console.aws.amazon.com/elasticache/> an AWS Management Console und öffnen Sie die ElastiCache Konsole.
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..
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</p>

Szenarien	Anstehende Operation	Neue Anfrage	Ergebnisse
			0003 gestellt, werden keine neuen Knoten angelegt und der Knoten 0003 wird gelöscht.
Szenario 4	Erstellen	Erstellen	<p>Der neue Erstellungsauftrag wird dem ausstehenden Erstellungsauftrag hinzugefügt.</p> <p>Wenn z. B. eine ausstehende Anforderung zur Erstellung von zwei Knoten besteht und eine neue Anforderung zur Erstellung von drei Knoten ausgegeben wird, wird die neue Anforderung der ausstehenden Anforderung hinzugefügt und es werden fünf Knoten erstellt.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Wenn der neue Erstellungsauftrag auf Sofort anwenden – Ja eingestellt ist, werden alle Erstellungsaufträge sofort ausgeführt. Wenn der neue Erstellungsauftrag auf Sofort anwenden – Nein eingestellt ist, sind alle Erstellungsaufträge in der Schwebe.</p> </div>

Um festzustellen, welche Vorgänge noch nicht abgeschlossen sind, wählen Sie die Registerkarte Beschreibung und prüfen Sie, wie viele ausstehende Erstellungen oder Löschungen angezeigt werden. Sie können nicht sowohl ausstehende Erstellungen als auch ausstehende Löschungen haben.

7. Wählen Sie die Schaltfläche Add.

Nach kurzer Zeit sollten die neuen Knoten mit dem Status creating in der Knotenliste erscheinen. Wenn dies nicht der Fall ist, aktualisieren Sie die Browserseite. Sobald sich der Status des Knotens in verfügbar ändert, kann der neue Knoten verwendet werden.

Verwenden Sie den AWS CLI

Verwenden Sie den AWS CLI Vorgang mit den folgenden Parametern AWS CLI, um Knoten zu `modify-cache-cluster` einem Cluster hinzuzufügen:

- `--cache-cluster-id` Die ID des Cache-Clusters, zu dem Sie Knoten hinzufügen möchten.
- `--num-cache-nodes` Der Parameter `--num-cache-nodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `--num-cache-nodes` größer als die aktuelle Anzahl von Knoten in diesem Cluster sein. Wenn dieser Wert kleiner als die aktuelle Anzahl von Knoten ist, ElastiCache erwartet, dass der Parameter `cache-node-ids-to-remove` und eine Liste von Knoten aus dem Cluster entfernt werden. Weitere Informationen finden Sie unter [Verwendung der AWS CLI](#).
- `--apply-immediately` oder `--no-apply-immediately` gibt an, ob diese Knoten umgehend bzw. während des nächsten Wartungsfensters hinzugefügt werden sollen.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 5 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 5 ^  
  --apply-immediately
```

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

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"    }  
  }  
}
```

```
    },
    "CacheClusterId": "my-cluster",
    "PreferredAvailabilityZone": "us-west-2b",
    "ConfigurationEndpoint": {
        "Port": 11211,
        "Address": "rlh-mem000.7alc7bf-example.cfg.usw2.cache.amazonaws.com"
    },
    "CacheSecurityGroups": [],
    "CacheClusterCreateTime": "2016-09-21T16:28:28.973Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "modifying",
    "NumCacheNodes": 2,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "SecurityGroups": [
        {
            "Status": "active",
            "SecurityGroupId": "sg-dbe93fa2"
        }
    ],
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {
        "NumCacheNodes": 5
    },
    "PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
    "CacheNodeType": "cache.m3.medium",
}
}
```

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

Die ElastiCache API verwenden

Um Knoten zu einem Cluster hinzuzufügen (ElastiCache API)

- Rufen Sie die API-Operation `ModifyCacheCluster` mit folgenden Parametern auf:
 - `CacheClusterId` Die ID des Clusters, zu dem Sie Knoten hinzufügen möchten.
 - `NumCacheNodes` Der Parameter `NumCacheNodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen. Damit diesem Cluster Knoten hinzugefügt werden, muss `NumCacheNodes` größer als die aktuelle Anzahl von Knoten

in diesem Cluster sein. Wenn dieser Wert unter der aktuellen Anzahl von Knoten liegt, wird ElastiCache erwartet, dass der Parameter `CacheNodeIdsToRemove` mit einer Liste von Knoten aus dem Cluster entfernt wird (siehe [Verwenden der ElastiCache-API](#)).

- `ApplyImmediately` Gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters hinzugefügt werden sollen.
- `Region` Gibt die AWS Region des Clusters an, zu der Sie Knoten hinzufügen möchten.

Das folgende Beispiel zeigt einen Aufruf zum Hinzufügen von Knoten zu einem Cluster.

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=true  
  &NumCacheNodes=5  
&CacheClusterId=my-cluster  
&Region=us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter ElastiCache API-Thema [ModifyCacheCluster](#).

Entfernen von Knoten aus einem Cluster

Jedes Mal, wenn Sie die Anzahl der Knoten in einem Memcached-Cluster ändern, müssen Sie zumindest einen Teil Ihres Keyspace neu zuordnen, sodass er dem richtigen Knoten zugeordnet ist. Detaillierte Informationen zum Lastausgleich eines Memcached-Clusters finden Sie unter [Konfigurieren Ihres ElastiCache-Clients für ein effizientes Load Balancing](#).

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

Verwendung der 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).

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 data-bbox="727 430 1507 842" 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 der 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 `modify-cache-cluster` wie im folgenden Beispiel die zu entfernenden Knoten auf.

Um über die Befehlszeilenschnittstelle Knoten aus einem Cluster zu entfernen, verwenden Sie den Befehl `modify-cache-cluster` mit den folgenden Parametern:

- `--cache-cluster-id` Die ID des Cache-Clusters, aus dem Sie die Knoten entfernen möchten.
- `--num-cache-nodes` Der Parameter `--num-cache-nodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- `--cache-node-ids-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.

Das folgende Beispiel entfernt umgehend Knoten 0001 aus dem Cluster my-cluster.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --num-cache-nodes 2 \  
  --cache-node-ids-to-remove 0001 \  
  --region us-east-2 \  
  --apply-immediately
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --num-cache-nodes 2 ^  
  --cache-node-ids-to-remove 0001 ^  
  --region us-east-2 ^  
  --apply-immediately
```

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

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",  
    "PreferredAvailabilityZone": "us-east-2b",  
    "ConfigurationEndpoint": {  
      "Port": 11211,  
      "Address": "rlh-mem000.7ef-example.cfg.usw2.cache.amazonaws.com"  
    },  
  },  
}
```

```
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-09-21T16:28:28.973Z", 9dcv5r
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "modifying",
"NumCacheNodes": 3,
"ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
"SecurityGroups": [
  {
    "Status": "active",
    "SecurityGroupId": "sg-dbe93fa2"
  }
],
"CacheSubnetGroupName": "default",
"EngineVersion": "1.4.24",
"PendingModifiedValues": {
  "NumCacheNodes": 2,
  "CacheNodeIdsToRemove": [
    "0001"
  ]
},
"PreferredMaintenanceWindow": "sat:09:00-sat:10:00",
"CacheNodeType": "cache.m3.medium",
}
}
```

Weitere Informationen finden Sie in den AWS CLI-Themen [describe-cache-cluster](#) und [modify-cache-cluster](#).

Verwenden der ElastiCache-API

Um Knoten mit der ElastiCache-API zu entfernen, rufen Sie den `ModifyCacheCluster`-API-Vorgang mit der Cache-Cluster-ID und einer Liste der zu entfernenden Knoten auf wie gezeigt:

- `CacheClusterId` Die ID des Cache-Clusters, aus dem Sie die Knoten entfernen möchten.
- `NumCacheNodes` Der Parameter `NumCacheNodes` gibt die Anzahl der Knoten an, die sich nach der Übernahme der Änderung in diesem Cluster befinden sollen.
- `CacheNodeIdsToRemove.member.n` Die Liste der Knoten-IDs, die Sie aus dem Cluster entfernen möchten.
 - `CacheNodeIdsToRemove.member.1=0004`

- `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` gibt an, ob diese Knoten umgehend oder während des nächsten Wartungsfensters entfernt werden sollen.
- `Region` Gibt die AWS-Region des Clusters an, aus dem Sie einen Knoten entfernen möchten.

Das folgende Beispiel entfernt umgehend Knoten 0004 und 0005 aus dem Cluster my-cluster.

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &CacheClusterId=my-cluster  
  &ApplyImmediately=true  
  &CacheNodeIdsToRemove.member.1=0004  
  &CacheNodeIdsToRemove.member.2=0005  
  &NumCacheNodes=3  
  &Region us-east-2  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie im Thema ElastiCache-API [ModifyCacheCluster](#).

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.

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 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 der 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

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

Wenn Sie Ihre ElastiCache Instance in einer Amazon Virtual Private Cloud (AmazonVPC) gestartet haben, können Sie von einer ElastiCache EC2 Amazon-Instance in demselben Amazon aus auf Ihre Instance zugreifenVPC. Oder Sie können mithilfe von VPC Peering von einem Amazon EC2 in einem anderen Amazon VPC auf Ihre ElastiCache Instance zugreifen.

Wenn Sie Ihre ElastiCache Instance in EC2 Classic gestartet haben, erlauben Sie der EC2 Instance den Zugriff auf Ihren Cluster, indem Sie der mit der Instance verknüpften EC2 Amazon-Sicherheitsgruppe Zugriff auf Ihre Cache-Sicherheitsgruppe gewähren. Standardmäßig ist der Zugriff auf einen Cluster auf das Konto beschränkt, mit dem der Cluster gestartet wurde.

Themen

- [Zugriff auf Ihren Cluster oder die erteilen](#)

Zugriff auf Ihren Cluster oder die erteilen

Sie haben Ihren Cluster gestartet in EC2 - VPC

Wenn Sie Ihren Cluster in einer Amazon Virtual Private Cloud (AmazonVPC) gestartet haben, können Sie nur von einer EC2 Amazon-Instance aus, die in demselben Amazon läuft, eine Verbindung zu Ihrem ElastiCache Cluster herstellenVPC. In diesem Fall müssen Sie Netzwerkzugang zum Cluster gewähren.


Note

Wenn Sie Local Zones verwenden, vergewissern Sie sich, dass Sie sie aktiviert haben. Weitere Informationen finden Sie unter [Local Zones aktivieren](#). Auf diese Weise VPC VPC wird Ihr Subnetz auf diese lokale Zone ausgedehnt und Sie behandeln das Subnetz wie jedes andere Subnetz in jeder anderen Availability Zone, und die relevanten Gateways, Routing-Tabellen und andere Überlegungen zu Sicherheitsgruppen werden automatisch angepasst.

So gewähren Sie Netzwerkzugang von einer VPC Amazon-Sicherheitsgruppe zu einem Cluster

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.

2. Wählen Sie in der Navigationsleiste unter Network & Security die Option Security Groups aus.
3. Wählen Sie aus der Liste der Sicherheitsgruppen die Sicherheitsgruppe für Ihr Amazon aus VPC. Sofern Sie keine Sicherheitsgruppe zur ElastiCache Verwendung erstellt haben, wird diese Sicherheitsgruppe als Standard bezeichnet.
4. Wählen Sie die Registerkarte Inbound und verfahren Sie dann wie folgt:
 - a. Wählen Sie Edit aus.
 - b. Wählen Sie Regel hinzufügen aus.
 - c. Wählen Sie in der Spalte Typ die Option Benutzerdefinierte TCP Regel aus.
 - d. Geben Sie in das Feld Port Range die Portnummer Ihres Clusterknotens ein. Diese Nummer muss mit der Nummer übereinstimmen, die Sie beim Starten des Clusters angegeben haben. Der Standardport für Memcached ist **11211** ist. **6379**
 - e. Wählen Sie im Feld Quelle die Option Anywhere mit dem Portbereich (0.0.0.0/0) aus, sodass jede EC2 Amazon-Instance, die Sie in Ihrem Amazon starten, eine Verbindung zu Ihren VPC Knoten herstellen kann. ElastiCache

 **Important**

Wenn Sie den ElastiCache Cluster für 0.0.0.0/0 öffnen, wird der Cluster nicht dem Internet zugänglich gemacht, da er keine öffentliche IP-Adresse hat und daher nicht von außerhalb des darauf zugegriffen werden kann. VPC Die Standardsicherheitsgruppe kann jedoch auf andere EC2 Amazon-Instances im Kundenkonto angewendet werden, und diese Instances können eine öffentliche IP-Adresse haben. Wenn diese Instances zufälligerweise eine Aktion auf dem Standardport ausführen, dann könnte dieser Service unbeabsichtigt zugänglich gemacht werden. Daher empfehlen wir, eine VPC Sicherheitsgruppe zu erstellen, die ausschließlich von verwendet wird ElastiCache. Weitere Informationen finden Sie unter [Benutzerdefinierte Sicherheitsgruppen](#).

- f. Wählen Sie Save (Speichern) aus.

Wenn Sie eine EC2 Amazon-Instance in Ihrem Amazon starten VPC, kann diese Instance eine Verbindung zu Ihrem ElastiCache Cluster herstellen.

Zugriff auf ElastiCache Ressourcen von außen AWS

Amazon ElastiCache ist ein AWS Service, der einen cloudbasierten In-Memory-Key-Value-Speicher bereitstellt. Der Service ist so konzipiert, dass er ausschließlich von innen abgerufen werden kann. Wenn der ElastiCache Cluster jedoch in einer VPC gehostet wird, können Sie eine Network Address Translation (NAT) -Instanz verwenden, um den Zugriff von außen zu ermöglichen.

Voraussetzungen

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

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

Überlegungen

Die folgenden Überlegungen sollten berücksichtigt werden, wenn Sie von außen ElastiCache auf Ihre ElastiCache Ressourcen zugreifen.

- Clients stellen eine Verbindung zum EIP und zum Cache-Port der NAT Instance her. Die Portweiterleitung auf der NAT Instance leitet den Datenverkehr an den entsprechenden Cache-Cluster-Knoten weiter.
- Wenn ein Cluster-Knoten hinzugefügt oder ersetzt wird, müssen die iptables-Regeln entsprechend aktualisiert werden.

Einschränkungen

Dieser Ansatz sollte nur zu Test- und Entwicklungszwecken verwendet werden. Er wird aufgrund der folgenden Einschränkungen nicht für die Produktion empfohlen:

- Die NAT Instanz fungiert als Proxy zwischen Clients und mehreren Clustern. Ein zusätzlicher Proxy beeinträchtigt die Leistung des Cache-Clusters. Die Auswirkung nimmt mit der Anzahl der Cache-Cluster zu, auf die Sie über die NAT Instance zugreifen.
- Der Datenverkehr von den Clients zur NAT Instance ist unverschlüsselt. Daher sollten Sie vermeiden, sensible Daten über die NAT Instance zu senden.
- Die NAT Instanz erhöht den Mehraufwand für die Wartung einer anderen Instanz.
- Die NAT Instanz dient als zentrale Fehlerquelle. Informationen zur Einrichtung von Hochverfügbarkeit finden Sie unter [Hochverfügbarkeit NAT für VPC NAT Amazon-Instances: Ein Beispiel](#). VPC

Wie greife ich von außen auf ElastiCache Ressourcen zu AWS

Das folgende Verfahren zeigt, wie Sie mithilfe einer NAT Instanz eine Verbindung zu Ihren ElastiCache Ressourcen herstellen.

Diese Schritte setzen Folgendes voraus:

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

Als Nächstes benötigen Sie NAT in die entgegengesetzte Richtung:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

Sie müssen auch die IP-Weiterleitung aktivieren, die standardmäßig deaktiviert ist:

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf  
sudo sysctl --system
```

- Ihr Zugriff auf einen Memcached-Cluster erfolgt mit:
 - IP-Adresse – 10.0.1.230
 - Memcached-Standardport – 11211
 - Sicherheitsgruppe – *10\0\0\0.55*
- Die IP-Adresse Ihres vertrauenswürdigen Clients lautet 198.51.100.27.
- Ihre NAT Instance hat die Elastic IP-Adresse 203.0.113.73.

- Ihre NAT Instance hat die Sicherheitsgruppe sg-ce56b7a9.

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

1. Erstellen Sie eine NAT Instanz im selben VPC wie Ihr Cache-Cluster, jedoch in einem öffentlichen Subnetz.

Standardmäßig startet der VPC Assistent den Knotentyp cache.m1.small. Wählen Sie eine Knotengröße aus, die Ihren Anforderungen entspricht. Sie müssen verwenden EC2 NATAMI, um von außen darauf zugreifen zu können. ElastiCache AWS

Informationen zum Erstellen einer NAT Instanz finden Sie im AWS VPC Benutzerhandbuch unter [NATInstanzen](#).

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

Die NAT Instanz-Sicherheitsgruppe und die Clusterinstanz sollten die folgenden Regeln haben:

- Zwei Regeln für eingehenden Datenverkehr
 - Eine, die TCP Verbindungen von vertrauenswürdigen Clients zu jedem Cache-Port ermöglicht, der von der NAT Instanz weitergeleitet wird (11211 — 11213).
 - Eine zweite, um den SSH Zugriff auf vertrauenswürdige Clients zu ermöglichen.

NATInstanz-Sicherheitsgruppe — Regeln für eingehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Benutzerdefinierte Regel TCP	TCP	11211 –11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

- Eine ausgehende Regel, die TCP Verbindungen zum Cache-Port (11211) zulässt.

NATInstanz-Sicherheitsgruppe — ausgehende Regel

Typ	Protocol (Protokoll)	Port-Bereich	Bestimmungsort
Benutzerdefinierte Regel TCP	TCP	11211	sg-ce56b7a9 (Sicherhe

Typ	Protocol (Protokoll)	Port-Bereich	Bestimmungsort
			itsgruppe der Cluster-Instance)

- Eine eingehende Regel für die Sicherheitsgruppe des Clusters, die TCP Verbindungen von der NAT Instance zum Cache-Port (11211) zulässt.

Sicherheitsgruppe der Cluster-Instance – Regel für eingehenden Datenverkehr

Typ	Protocol (Protokoll)	Port-Bereich	Quelle
Benutzerdefinierte Regel TCP	TCP	11211	sg-bd56b7 da (Sicherheitsgruppe) NAT

3. Validieren Sie die Regeln.

- Stellen Sie sicher, dass der vertrauenswürdige Client Zugriff SSH auf die NAT Instanz hat.
- Vergewissern Sie sich, dass der vertrauenswürdige Client von der NAT Instanz aus eine Verbindung zum Cluster herstellen kann.

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

Für jeden Knoten im Cluster muss der NAT Tabelle eine iptables-Regel hinzugefügt werden, um den Cache-Port von der NAT Instanz an den Clusterknoten weiterzuleiten. Ein Beispiel könnte folgendermaßen aussehen:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to 10.0.1.230:11211
```

Die Portnummer muss für jeden Knoten im Cluster eindeutig sein. Wenn Sie z. B. mit einem aus drei Knoten bestehenden Memcached-Cluster unter Verwendung der Ports 11211 – 11213 arbeiten, sehen die Regeln wie folgt aus:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to 10.0.1.230:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to 10.0.1.231:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to 10.0.1.232:11211
```

5. Bestätigen Sie, dass der vertrauenswürdige Client eine Verbindung mit dem Cluster herstellen kann.

Der vertrauenswürdige Client sollte eine Verbindung zu dem mit der NAT Instanz EIP verknüpfen und dem Cluster-Port herstellen, der dem entsprechenden Clusterknoten entspricht. Die Verbindungszeichenfolge für PHP könnte beispielsweise wie folgt aussehen:

```
$memcached->connect( '203.0.113.73', 11211 );  
$memcached->connect( '203.0.113.73', 11212 );  
$memcached->connect( '203.0.113.73', 11213 );
```

Zum Überprüfen der Verbindung kann auch ein Telnet-Client verwendet werden. Beispielsweise:

```
telnet 203.0.113.73 11211  
telnet 203.0.113.73 11212  
telnet 203.0.113.73 11213
```

6. 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](#)
- [NATInstanzen](#)
- [Konfiguration von ElastiCache Clients](#)
- [Hochverfügbarkeit für VPC NAT Amazon-Instances: Ein Beispiel](#)

Ermitteln von Verbindungsendpunkten

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

Zu verwendende Endpunkte

Für ElastiCache serverlosen Cache mit Memcached rufen Sie einfach den Cluster-Endpunkt DNS und den Port von der Konsole ab.

Verwenden Sie von der aus den `describe-serverless-caches` Befehl AWS CLI, um die Endpunktinformationen abzurufen.

Linux

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

Windows

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

Die Ausgabe der obigen Operation sollte ungefähr so aussehen (JSONFormat):

```
{
  "ServerlessCaches": [
    {
      "ServerlessCacheName": "serverless-memcached",
      "Description": "test",
      "CreateTime": 1697659642.136,
      "Status": "available",
      "Engine": "memcached",
      "MajorEngineVersion": "1.6",
      "FullEngineVersion": "21",
      "SecurityGroupIds": [
        "sg-083eda453e1e51310"
      ],
      "Endpoint": {
        "Address": "serverless-memcached-01.amazonaws.com",
        "Port": 11211
      },
      "ARN": "<the ARN>",
    }
  ]
}
```

```
        "SubnetIds": [
            "subnet-0cf759df15bd4dc65",
            "subnet-09e1307e8f1560d17"
        ],
        "SnapshotRetentionLimit": 0,
        "DailySnapshotTime": "03:00"
    }
]
}
```

Instance-basierter Memcached-Cluster – Wenn Sie Automatic Discovery verwenden, können Sie Ihren Memcached-Client über den Konfigurationsendpunkt des Clusters konfigurieren. Dies bedeutet, dass Sie einen Client verwenden müssen, der Automatic Discovery unterstützt.

Wenn Sie Automatic Discovery nicht verwenden, müssen Sie den Client so konfigurieren, dass er die einzelnen Knotenendpunkte für Lese- und Schreibvorgänge verwendet. Auch müssen Sie sie beim Hinzufügen und Entfernen von Knoten verfolgen.

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

Finden der Endpunkte eines Clusters (Konsole)

Alle Memcached-Endpunkte sind Lese-/Schreib-Endpunkte. Zum Herstellen einer Verbindung mit Knoten in einem Memcached-Cluster kann Ihre Anwendung entweder die Endpunkte für jeden Knoten oder den Konfigurationsendpunkt des Clusters zusammen mit Automatic Discovery verwenden. Um Automatic Discovery verwenden zu können, benötigen Sie einen Client, der Automatic Discovery unterstützt.

Wenn Automatic Discovery verwendet wird, stellt Ihre Client-Anwendung über den Konfigurationsendpunkt eine Verbindung mit Ihrem Memcached-Cluster her. Während Sie Ihren Cluster durch Hinzufügen oder Entfernen von Knoten skalieren, „kennt“ Ihre Anwendung automatisch alle Knoten im Cluster und kann zu beliebigen von ihnen eine Verbindung herstellen. Ohne Automatic Discovery wäre das Ihre Aufgabe oder Sie müssten die Endpunkte Ihrer Anwendung jedes Mal manuell aktualisieren, wenn Sie einen Knoten hinzufügen oder löschen würden.

Wenn Sie einen Endpunkt kopieren möchten, wählen Sie das Kopiersymbol direkt vor der Endpunktadresse aus. Informationen zur Verwendung des Endpunkts zur Verbindung mit einem Knoten finden Sie unter [Mit Knoten verbinden](#).

Die Konfiguration und die Knotenendpunkte sehen sehr ähnlich aus. Es folgen die mit Fettdruck hervorgehobenen Unterschiede.

```
myclustername.xxxxxx.cfg.usw2.cache.amazonaws.com:port # configuration endpoint  
contains "cfg"  
myclustername.xxxxxx.0001.usw2.cache.amazonaws.com:port # node endpoint for node 0001
```

Important

Wenn Sie einen Konfigurationsendpunkt CNAME für Ihren Memcached erstellen möchten, damit Ihr Automatic Discovery Client ihn CNAME als Konfigurationsendpunkt erkennt, müssen Sie ihn in den einschließen. `.cfg.CNAME`

Suchen von Endpunkten (AWS CLI)

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

Themen

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

Suchen von Endpunkten für Knoten und Cluster (AWS CLI)

Sie können den verwenden AWS CLI , um die Endpunkte für einen Cluster und seine Knoten mit dem `describe-cache-clusters` Befehl zu ermitteln. Für Memcached-Cluster gibt der Befehl den Konfigurationsendpunkt zurück. Wenn Sie den optionalen Parameter `--show-cache-node-info` einschließen, gibt der Befehl auch die Endpunkte der einzelnen Knoten im Cluster zurück.

Example

Der folgende Befehl ruft den Konfigurationsendpunkt (`ConfigurationEndpoint`) und einzelne Knotenendpunkte (`Endpoint`) für den Memcached-Cluster `mycluster` ab.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-clusters \  
  --cache-cluster-id mycluster \  
  --show-cache-node-info
```

Für Windows:

```
aws elasticache describe-cache-clusters ^  
  --cache-cluster-id mycluster ^  
  --show-cache-node-info
```

Die Ausgabe des obigen Vorgangs sollte ungefähr so aussehen (JSONFormat).

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {  
          "CacheNodeId": "0001",
```

```
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  },
  {
    "CacheNodeId": "0002",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  },
  {
    "CacheNodeId": "0003",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  }
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "us-west-2b",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.amazonaws.com"
},
"CacheSecurityGroups": [],
```

```
    "CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
    "AutoMinorVersionUpgrade": true,
    "CacheClusterStatus": "available",
    "NumCacheNodes": 3,
    "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
    "CacheSubnetGroupName": "default",
    "EngineVersion": "1.4.24",
    "PendingModifiedValues": {},
    "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
    "CacheNodeType": "cache.m4.large",
  }
]
}
```

Important

Wenn Sie sich dafür entscheiden, einen Konfigurationsendpunkt CNAME für Ihren Memcached zu erstellen, damit Ihr Auto Discovery-Client ihn CNAME als Konfigurationsendpunkt erkennt, müssen Sie ihn `.cfg.` in den aufnehmen. CNAME Zum Beispiel `mycluster.cfg.local` in Ihrer `php.ini`-Datei für den Parameter `session.save_path`.

Weitere Informationen finden Sie im Thema [describe-cache-clusters](#).

Endpunkte finden () ElastiCache API

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

Themen

- [Endpunkte für Knoten und Cluster finden \(\) ElastiCache API](#)

Endpunkte für Knoten und Cluster finden () ElastiCache API

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

Example

Der folgende Befehl ruft den Konfigurationsendpunkt (`ConfigurationEndpoint`) und einzelne Knotenendpunkte (`Endpoint`) für den Memcached-Cluster `mycluster` ab.

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

Important

Wenn Sie sich dafür entscheiden, einen Konfigurationsendpunkt CNAME für Ihren Memcached zu erstellen, damit Ihr Auto Discovery-Client ihn CNAME als Konfigurationsendpunkt erkennt, müssen Sie ihn `.cfg.` in den aufnehmen. CNAME Zum Beispiel `mycluster.cfg.local` in Ihrer `php.ini`-Datei für den Parameter `session.save_path`.

Knoten verwalten

Ein Knoten ist der kleinste Baustein einer ElastiCache Amazon-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 ausgewählt wurde. Jeder Knoten besitzt einen eigenen Domain Name Service (DNS)-Namen und Port. Es werden mehrere ElastiCache Knotentypen unterstützt, die jeweils unterschiedliche Mengen an Arbeitsspeicher und Rechenleistung aufweisen.

Eine ausführliche Erörterung der zu verwendenden Knotengröße finden Sie unter [Auswahl der Memcached-Knotengröße](#).

Themen

- [ElastiCache Knotenstatus anzeigen](#)
- [Mit Knoten verbinden](#)
- [Unterstützte Knotentypen](#)
- [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 \(Memcached\)](#)
- [Ermitteln von Verbindungsendpunkten](#)

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, den Knoten zu ändern, geändert.
updating	Berechnet	<p>Ein Aktualisierungsstatus gibt an, dass eine oder mehrere der folgenden Bedingungen für den ElastiCache Amazon-Knoten zutreffen:</p> <ul style="list-style-type: none">• Der ElastiCache Knoten wird im Rahmen des Service-Updates gepatcht. Weitere Informationen zu den Service-Updates finden Sie auf der Hilfeseite Amazon ElastiCache Managed Maintenance and Service Updates.• Die VPC-Sicherheitsgruppen werden für den ElastiCache Cluster aktualisiert.

Typ	Berechnet	Beschreibung
		<ul style="list-style-type: none">• Der ElastiCache Cluster wird hoch- oder herunterskaliert.• Die Konfigurationen für die Protokollzustellung werden für den ElastiCache Cluster geändert.• Ein Löschvorgang für den ElastiCache Knoten steht noch aus.• Das ElastiCache (Redis OSS) -Passwort wird mit aktualisiert/rotiert. AWS Secrets Manager
rebooting cache cluster nodes	Berechnet	Der ElastiCache Knoten wird aufgrund einer Kundenanfrage oder eines ElastiCache Amazon-Prozesses, der einen Neustart des Knotens erfordert , neu gestartet.

Typ	Berechnet	Beschreibung
<code>incompatible_parameters</code>	Nicht berechnet	<p>Amazon ElastiCache kann den Knoten nicht starten, da die in der Parametergruppe des Knotens angegebenen Parameter nicht mit dem Knoten kompatibel sind. Machen Sie entweder die letzten Änderungen an den Parametern rückgängig oder stellen Sie ihre Kompatibilität mit dem Knoten her, um wieder auf den Knoten zugreifen zu können. Weitere Informationen zu den inkompatiblen Parametern finden Sie in der Ereignisliste für den ElastiCache Knoten.</p>
<code>incompatible_network</code>	Nicht berechnet	<p>Ein inkompatibler Netzwerkstatus bedeutet, dass eine oder mehrere der folgenden Bedingungen auf den Amazon-Knoten zutreffen: ElastiCache</p> <ul style="list-style-type: none">• In dem Subnetz, in dem der ElastiCache Knoten gestartet wurde, sind keine IP-Adressen verfügbar.• Das in der Subnetzgruppe erwähnte ElastiCache Subnetz ist in der Amazon Virtual Private Cloud (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 Knotens ElastiCache (Redis OSS).

Den ElastiCache Knotenstatus mit der Konsole anzeigen

So zeigen Sie den Status eines ElastiCache Nodes mit der Konsole an:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich Redis OSS Clusters oder Memcached Clusters aus. Die Seite „Caches“ mit der Liste der Knoten wird angezeigt. ElastiCache Für die einzelnen Knoten wird der Statuswert angezeigt.
3. Sie können dann zur Registerkarte Service Updates für den Cache navigieren, um die Liste der Service-Updates anzuzeigen, die für den Cache gelten.

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.

Mit Knoten verbinden

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

- [Finden der Endpunkte eines Clusters \(Konsole\)](#)
- [Suchen von Endpunkten \(AWS CLI\)](#)
- [Endpunkte finden \(\) ElastiCache API](#)

Im folgenden Beispiel verwenden Sie das telnet-Dienstprogramm, um eine Verbindung mit einem Knoten herzustellen, der Memcached ausführt.

Note

Weitere Informationen über Memcached und verfügbare Memcached-Befehle finden Sie auf der [Memcached](#)-Website.

So stellen Sie eine Verbindung mit einem Knoten mithilfe von telnet her

1. Stellen Sie eine Verbindung zu Ihrer Amazon-EC2-Instance her, indem Sie das Verbindungsprogramm Ihrer Wahl verwenden.

Note

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

2. Laden Sie das Telnet-Dienstprogramm herunter und installieren Sie es auf Ihrer Amazon-EC2-Instance. Geben Sie an der Eingabeaufforderung Ihrer Amazon-EC2-Instance den folgenden Befehl ein und geben Sie an der Eingabeaufforderung y ein.

```
sudo yum install telnet
```

Die Ausgabe sieht in etwa folgendermaßen aus.


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

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB      00:00

...(output omitted)...

Complete!
```

3. Geben Sie an der Eingabeaufforderung Ihrer Amazon-EC2-Instance den folgenden Befehl ein, wobei Sie den Endpunkt Ihres Knotens durch den in diesem Beispiel gezeigten ersetzen.

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

Die Ausgabe sieht in etwa folgendermaßen aus.

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

4. Prüfen Sie die Verbindung mithilfe der Memcached-Befehle.

Sie sind nun mit einem Knoten verbunden und können Memcached-Befehle ausführen. Im Folgenden wird ein Beispiel gezeigt.

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

```
END
```

```
>
```

Unterstützte Knotentypen

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

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](#).

- Allgemeine Zwecke:
- Aktuelle Generation:

M6g-Knotentypen (nur für die Memcached-Engine ab Version 1.5.16 verfügbar).

cache.m6g.large, cache.m6g.xlarge, cache.m6g.2xlarge, cache.m6g.4xlarge, cache.m6g.8xlarge, cache.m6g.12xlarge, cache.m6g.16xlarge

Note

Informationen zur Verfügbarkeit in der Region finden Sie unter [Unterstützte Knotentypen nach AWS -Region](#).

M5-Knotentypen: cache.m5.large, cache.m5.xlarge, cache.m5.2xlarge, cache.m5.4xlarge, cache.m5.12xlarge, cache.m5.24xlarge

M4-Knotentypen: cache.m4.large, cache.m4.xlarge, cache.m4.2xlarge, cache.m4.4xlarge, cache.m4.10xlarge

T4g-Knotentypen (nur für die Memcached-Engine ab Version 1.5.16 verfügbar).

cache.t4g.micro, cache.t4g.small, cache.t4g.medium

T3-Knotentypen: cache.t3.micro, cache.t3.small, cache.t3.medium

T2-Knotentypen: cache.t2.micro, cache.t2.small, cache.t2.medium

- Vorherige Generation (nicht empfohlen): Bestehende Cluster werden weiterhin unterstützt, aber die Erstellung neuer Cluster wird für diese Typen nicht unterstützt.)

T1-Knotentypen: `cache.t1.micro`

M1-Knotentypen: `cache.m1.small`, `cache.m1.medium`, `cache.m1.large`,
`cache.m1.xlarge`

M3-Knotentypen: `cache.m3.medium`, `cache.m3.large`, `cache.m3.xlarge`,
`cache.m3.2xlarge`


- Für Datenverarbeitung optimiert:
 - Vorherige Generation (nicht empfohlen):

C1-Knotentypen: `cache.c1.xlarge`

- RAM-optimiert:
 - Aktuelle Generation:

(R6g-Knotentypen sind nur für die Memcached-Engine ab Version 1.5.16 verfügbar.)

R6g-Knotentypen: `cache.r6g.large`, `cache.r6g.xlarge`, `cache.r6g.2xlarge`,
`cache.r6g.4xlarge`, `cache.r6g.8xlarge`, `cache.r6g.12xlarge`,
`cache.r6g.16xlarge`

 Note

Informationen zur Verfügbarkeit in der Region finden Sie unter [Unterstützte Knotentypen nach AWS -Region](#).

R5-Knotentypen: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`,
`cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

R4-Knotentypen: `cache.r4.large`, `cache.r4.xlarge`, `cache.r4.2xlarge`,
`cache.r4.4xlarge`, `cache.r4.8xlarge`, `cache.r4.16xlarge`

- Vorherige Generation (nicht empfohlen):

M2-Knotentypen: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

R3-Knotentypen: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`,
`cache.r3.4xlarge`, `cache.r3.8xlarge`

- Netzwerkoptimiert:

- Aktuelle Generation:

(C7gn-Knotentypen sind nur für die Memcached-Engine ab Version 1.6.6 verfügbar.)

C7gn-Knotentypen: `cache.c7gn.large`, `cache.c7gn.xlarge`, `cache.c7gn.2xlarge`,
`cache.c7gn.4xlarge`, `cache.c7gn.8xlarge`, `cache.c7gn.12xlarge`,
`cache.c7gn.16xlarge`

Aktuelle Generation

Die folgende Tabellen zeigen die Basisbandbreite und die Burst-Bandbreite für Instance-Typen, die den Mechanismus der Netzwerk-E/A-Gutschrift verwenden, um über ihre Basisbandbreite hinaus zu gehen.

Note

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

Allgemeines

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
<code>cache.m7g.large</code>		0,937	12,5
<code>cache.m7g.xlarge</code>		1,876	12,5
<code>cache.m7g.2xlarge</code>		3,75	15
<code>cache.m7g.4xlarge</code>		7,5	15

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.m7g.8xlarge		15	N/A
cache.m7g.12xlarge		22,5	N/A
cache.m7g.16xlarge		30	N/A
cache.m6g.large	1,5,16	0.75	10.0
cache.m6g.xlarge	1.5,16	1,25	10.0
dache.m6g.2xlarge	1,5,16	2.5	10.0
dache.m6g.4xlarge	1.5,16	5.0	10.0
dache.m6g.8xlarge	1.5,16	12	N/A
dache.m6g.12xlarge	1.5,16	20	N/A
dache.m6g.16xlarge	1.5,16	25	N/A
cache.m5.large	1.5,16	0.75	10.0
cache.m5.xlarge	1.5,16	1,25	10.0
cache.m5.2xlarge	1,5,16	2.5	10.0
cache.m5.4xlarge	1.5,16	5.0	10.0
cache.m5.12xlarge	1.5,16	N/A	N/A
cache.m5.24xlarge	1.5,16	N/A	N/A
cache.m4.large	1.5,16	0,45	1.2
cache.m4.xlarge	1,5,16	0.75	2.8
cache.m4.2xlarge	1.5,16	1,0	10.0

Instance-Typ	Unterstützte Memcached-Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.m4.4xlarge	1.5,16	2.0	10.0
cache.m4.10xlarge	1.5,16	5.0	10.0
cache.t4g.micro	1.5,16	0,064	5.0
cache.t4g.small	1,5,16	0,128	5.0
cache.t4g.medium	1,5,16	0,256	5.0
cache.t3.micro	1,5,16	0,064	5.0
cache.t3.small	1,5,16	0,128	5.0
cache.t3.medium	1,5,16	0,256	5.0
cache.t2.micro	1,5,16	0,064	1,024
cache.t2.small	1,5,16	0,128	1,024
cache.t2.medium	1,5,16	0,256	1,024

RAM-optimiert

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r7g.large		0,937	12,5
cache.r7g.xlarge		1,876	12,5
cache.r7g.2xlarge		3,75	15
cache.r7g.4xlarge		7,5	15
cache.r7g.8xlarge		15	N/A

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r7g.12xlarge		22,5	N/A
cache.r7g.16xlarge		30	N/A
cache.r6g.large	1,5,16	0.75	10.0
cache.r6g.xlarge	1.5,16	1,25	10.0
cache.r6g.2xlarge	1,5,16	2.5	10.0
cache.r6g.4xlarge	1.5,16	5.0	10.0
cache.r6g.8xlarge	1.5,16	12	N/A
cache.r6g.12xlarge	1.5,16	20	N/A
cache.r6g.16xlarge	1.5,16	25	N/A
cache.r5.large	1.5,16	0.75	10.0
cache.r5.xlarge	1.5,16	1,25	10.0
cache.r5.2xlarge	1,5,16	2.5	10.0
cache.r5.4xlarge	1.5,16	5.0	10.0
cache.r5.12xlarge	1.5,16	20	N/A
cache.r5.24xlarge	1.5,16	25	N/A
cache.r4.large	1.5,16	0.75	10.0
cache.r4.xlarge	1.5,16	1,25	10.0
cache.r4.2xlarge	1,5,16	2.5	10.0
cache.r4.4xlarge	1.5,16	5.0	10.0
cache.r4.8xlarge	1.5,16	12	N/A

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.r4.16xlarge	1.5,16	25	N/A

Netzwerkoptimiert

Instance-Typ	Unterstützte Mindestversion	Baseline-Bandbreite (Gbit/s)	Maximale Bandbreite (Gbit/s)
cache.c7gn.large	1.6.6	6,25	30
cache.c7gn.xlarge	1,6.6	12,5	40
cache.c7gn.2xlarge	1.6.6	25	50
cache.c7gn.4xlarge	1.6.6	50	N/A
cache.c7gn.8xlarge	1.6.6	100	N/A
cache.c7gn.12xlarge	1.6.6	150	N/A
cache.c7gn.16xlarge	1.6.6	200	N/A

Unterstützte Knotentypen nach AWS -Region

Die unterstützten Knotentypen können je nach AWS Region variieren. Weitere Informationen finden Sie unter [ElastiCacheAmazon-Preise](#).

Instances mit Spitzenlastleistung

Sie können in Amazon Allzweck-Burstable T4g-, T3-Standard- und T2-Standard-Cache-Knoten starten. ElastiCache Diese Knoten bieten 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.

ElastiCacheDie T4g-, T3- und T2-Knoten von Amazon sind standardmäßig konfiguriert und eignen sich für Workloads mit einer durchschnittlichen CPU-Auslastung, die durchweg unter der Basisleistung der Instance liegt. Um 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

Pro Stunde erworbenes CPU-Guthaben	Maximal verdiente Guthaben, die angesammelt werden können*	vCPUs	Basisleistung pro vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung
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.

Verwandte Informationen

- [ElastiCache Amazon-Produktmerkmale und -details](#)
- [Spezifische Parameter des Memcached-Knotentyps](#)

Ersetzen von Knoten

Amazon ElastiCache (Memcached) aktualisiert seine Flotte häufig, indem Patches und Upgrades nahtlos auf Instances angewendet werden. Von Zeit zu Zeit müssen wir jedoch Ihre ElastiCache (Memcached-) Nodes neu starten, um obligatorische Betriebssystem-Updates auf den zugrunde liegenden Host anzuwenden. Diese Ersetzungen sind erforderlich, um Upgrades anzuwenden, die die Sicherheit, Zuverlässigkeit und Betriebsleistung erhöhen.

Sie haben die Möglichkeit, diese Ersetzungen vor dem geplanten Knotenersetzungsfenster jederzeit selbst zu verwalten. Wenn Sie eine Ersetzung selbst verwalten, erhält Ihre Instance das Betriebssystem-Update, wenn Sie den Knoten neu starten, und der geplante Austausch des Knotens wird abgebrochen. Sie erhalten möglicherweise weiterhin Benachrichtigungen darüber, dass der Knotenaustausch stattfindet. Wenn Sie die erforderliche Wartung bereits manuell vorgenommen haben, können Sie diese Benachrichtigungen ignorieren.

Note

Von Amazon automatisch generierte Ersatz-Cache-Knoten ElastiCache können unterschiedliche IP-Adressen haben. Sie sind dafür verantwortlich, Ihre Anwendungskonfiguration zu überprüfen, um sicherzustellen, dass Ihre Cache-Knoten den entsprechenden IP-Adressen zugeordnet sind.

In der folgenden Liste sind die Maßnahmen aufgeführt, die Sie ergreifen können, ElastiCache wenn Sie den Austausch eines Ihrer Memcached-Knoten planen.

- Nichts tun — Wenn Sie nichts tun, wird der Knoten wie geplant ElastiCache ersetzt. Wenn der Knoten ElastiCache automatisch durch einen neuen Knoten ersetzt wird, ist der neue Knoten zunächst leer.
- Ändern Sie Ihr Wartungsfenster — Bei geplanten Wartungsereignissen erhalten Sie eine E-Mail oder eine Benachrichtigung von ElastiCache. Wenn Sie in diesen Fällen Ihr Wartungsfenster vor der geplanten Ersatzzeit ändern, wird Ihr Knoten zur neuen Uhrzeit ersetzt. Weitere Informationen finden Sie unter [Einen ElastiCache Cluster ändern](#).

Note

Die Möglichkeit, Ihr Ersatzfenster zu ändern, indem Sie Ihr Wartungsfenster verschieben, ist nur verfügbar, wenn die ElastiCache Benachrichtigung ein Wartungsfenster beinhaltet.

Wenn die Benachrichtigung kein Wartungsfenster enthält, können Sie Ihr Ersatzfenster nicht ändern.

Es ist beispielsweise Donnerstag, der 9. November, um 15:00 Uhr und das nächste Wartungsfenster ist am Freitag, dem 10. November, um 17:00 Uhr. Es folgen drei Szenarien mit den jeweiligen Ergebnissen:

- Sie ändern Ihr Wartungsfenster auf Freitag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben, aber vor dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Freitag, dem 10. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Samstag, 16:00 Uhr, nach den aktuellen Datums- und Uhrzeitangaben und nach dem nächsten geplanten Wartungsfenster. Dieser Knoten wird am Samstag, dem 11. November, um 16:00 Uhr ersetzt.
- Sie ändern Ihr Wartungsfenster auf Mittwoch, 16:00 Uhr, früher in der Woche, als die aktuellen Datums- und Uhrzeitangaben. Dieser Knoten wird am kommenden Mittwoch, dem 15. November, um 16:00 Uhr ersetzt.

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

- Manuelles Ersetzen des Knotens – Wenn Sie den Knoten vor dem nächsten Wartungsfenster ersetzen müssen, ersetzen Sie den Knoten manuell.

Wenn Sie den Knoten manuell ersetzen, werden Schlüssel umverteilt. Die Umverteilung verursacht Cache-Fehler.

So ersetzen Sie einen Memcached-Knoten manuell

1. Löschen Sie den für das Ersetzen geplanten Knoten. Anweisungen finden Sie unter [Entfernen von Knoten aus einem Cluster](#).
2. Fügen Sie dem Cluster einen neuen Knoten hinzu. Anweisungen finden Sie unter [Knoten zu einem Cluster hinzufügen](#).
3. Wenn Sie Auto Discovery in diesem Cluster nicht verwenden, rufen Sie Ihre Anwendung auf und tauschen Sie jede Instance des Endpunkts des alten mit dem Endpunkt des neuen Knotens aus.

Reservierte Knoten 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 die Option Memcached aus.
5. Treffen Sie zur Ermittlung der verfügbaren Angebote eine Auswahl für die folgenden Optionen:
 - Node Type
 - Laufzeit
 - Angebotstyp

Nachdem Sie Ihre Auswahl getroffen haben, werden die Kosten pro Knoten und die Gesamtkosten der ausgewählten Optionen unter Reservation details (Reservierungsdetails) angezeigt.

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

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

Weitere Informationen finden Sie 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 die Option Memcached aus.
5. Treffen Sie zur Ermittlung der verfügbaren Angebote eine Auswahl für die folgenden Optionen:
 - Node Type
 - Laufzeit
 - Angebotstyp
 - Eine optionale Reserved-Node-ID

Nachdem Sie Ihre Auswahl getroffen haben, werden die Kosten pro Knoten und die Gesamtknoten der ausgewählten Optionen unter Reservation details (Reservierungsdetails) angezeigt.

6. Klicken Sie auf Purchase (Kaufen).

Kauf eines reservierten Knotens (AWS CLI)

Das folgende Beispiel zeigt den Erwerb des bestimmten reservierten Clusters, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der Knoten-ID myreservationID.

Geben Sie den folgenden Befehl an einer Eingabeaufforderung ein:

Für Linux, macOS oder Unix:

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

Für Windows:

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

Daraufhin erhalten Sie ein Ergebnis, das dem hier dargestellten entspricht:

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

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

Kauf eines reservierten Knotens (ElastiCache API)

Das folgende Beispiel zeigt den Kauf des bestimmten reservierten Knotens, 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f, mit der reservierten Cluster-ID myreservationID.

Rufen Sie die `PurchaseReservedCacheNodesOffering`-Operation mit folgenden Parametern auf:

- `ReservedCacheNodesOfferingId` = 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
- `ReservedCacheNodeID` = myreservationID
- `CacheNodeCount` = 1

Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=PurchaseReservedCacheNodesOffering  
  &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  
  &ReservedCacheNodeID=myreservationID  
  &CacheNodeCount=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie 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",

"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 vorhandenen Cluster haben, die einen Knotentyp der vorherigen Generation verwenden, ElastiCache unterstützt die Erstellung neuer Cluster mit diesem Knotentyp nicht.

Aufgrund der begrenzten Anzahl von Node-Typen der vorherigen Generation können wir keinen erfolgreichen Ersatz garantieren, wenn ein Node in Ihrem(n) Cluster(n) fehlerhaft wird. In einem solchen Szenario kann die Verfügbarkeit Ihres Clusters negativ beeinflusst werden.

Wir empfehlen, dass Sie Ihre(n) Cluster auf einen neuen Knotentyp migrieren, um die Verfügbarkeit und Leistung zu verbessern. Einen empfohlenen Knotentyp für die Migration finden Sie unter [Upgrade-Pfade](#). Eine vollständige Liste der unterstützten Knotentypen und Knotentypen der vorherigen Generation finden Sie unter [Unterstützte Knotentypen](#). ElastiCache

Migrieren von Knoten auf einen Memcached-Cluster

Um ElastiCache (Memcached) zu einem anderen Knotentyp zu migrieren, müssen Sie einen neuen Cluster erstellen, der immer leer ist und von Ihrer Anwendung aufgefüllt werden kann.

So migrieren Sie Ihren ElastiCache (Memcached) Cluster-Knotentyp mithilfe der Konsole:
ElastiCache

- Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).
- Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Finden der Endpunkte eines Clusters \(Konsole\)](#).
- Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Löschen eines Clusters](#)

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 \(Memcached\)](#)
- [Markieren Ihrer ElastiCache-Ressourcen](#)
- [Verwendung von Amazon ElastiCache Well-Architected Lens](#)
- [Allgemeine Schritte und bewährte Methoden zur Fehlerbehebung](#)
- [Zusätzliche Schritte zur Fehlerbehebung](#)

Snapshot und Wiederherstellung

ElastiCache Amazon-Caches, auf denen Serverless Memcached 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 Serverless Memcached-Cache 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](#)
- [Planen automatischer Backups](#)
- [Erstellen manueller Backups](#)
- [Erstellen einer endgültigen Sicherung](#)
- [Beschreiben von Sicherungen](#)
- [Kopieren eines Backups](#)
- [Wiederherstellen aus einem Backup in einen neuen Cache](#)
- [Löschen einer Sicherung](#)
- [Markieren von Sicherungen](#)

Sicherungseinschränkungen

Berücksichtigen Sie die folgenden Einschränkungen bei der Planung und Erstellung von Sicherungen:

- Backup und Wiederherstellung werden nur für Caches unterstützt, die auf Redis OSS oder Serverless Memcached ausgeführt werden.
- Während eines zusammenhängenden Zeitraums von 24 Stunden können Sie nicht mehr als 24 manuelle Backups pro serverlosem Cache erstellen.
- Während des Backup-Vorgangs können Sie keine anderen API- oder CLI-Operationen im serverlosen Cache ausführen.

Planen automatischer Backups

Sie können automatische Backups für jeden Memcached Serverless Cache aktivieren. Wenn automatische Backups aktiviert sind, ElastiCache erstellt täglich eine Sicherungskopie des Caches. Es gibt keine Auswirkungen auf den Cache und die Änderung erfolgt sofort. Automatische Backups schützen vor Datenverlust. Bei einem Ausfall können Sie einen neuen Cache erstellen, indem Sie Ihre Daten aus dem aktuellen Backup wiederherstellen. Das Ergebnis ist ein warm gestarteter Cache, auf dem Ihre Daten vorab geladen und einsatzbereit sind. Weitere Informationen finden Sie unter [Wiederherstellen aus einem Backup in einen neuen Cache](#).

Wenn Sie automatische Backups planen, legen Sie fest, wann mit der Erstellung des Backups begonnen ElastiCache 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.

Sie können automatische Backups aktivieren oder deaktivieren, wenn Sie einen Memcached Serverless-Cache erstellen, indem Sie die ElastiCache Konsole, die oder die API AWS CLI verwenden. ElastiCache Aktivieren Sie dazu im Abschnitt Erweiterte Memcached-Einstellungen das Kästchen Automatische Backups aktivieren.

Erstellen manueller Backups

Zusätzlich zu automatischen Sicherungen können Sie jederzeit eine manuelle Sicherung erstellen. Im Gegensatz zu automatischen Sicherungen, die nach einem angegebenen Aufbewahrungszeitraum automatisch gelöscht werden, ist für manuelle Sicherungen kein Aufbewahrungszeitraum vorhanden, nachdem sie automatisch gelöscht werden. Selbst wenn Sie den Cache löschen, werden alle manuellen Backups aus diesem Cache beibehalten. Wenn Sie eine manuelle Sicherung nicht mehr aufbewahren möchten, müssen Sie sie selbst explizit löschen.

Eine manuelle Sicherung kann nicht nur direkt, sondern auch auf eine der folgenden Arten erstellt werden:

- [Kopieren eines Backups](#). Es spielt keine Rolle, ob die Quellsicherung automatisch oder manuell erstellt wurde.
- [Erstellen einer endgültigen Sicherung](#). Erstellen Sie ein Backup unmittelbar vor dem Löschen eines Clusters oder Knotens.

Sie können 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 Memcached Caches aus.
3. Wählen Sie das Feld links neben dem Namen des Caches aus, den Sie sichern möchten.
4. Wählen Sie Backup.
5. Geben Sie im Dialog Create Backup einen Namen für Ihre Sicherung im Feld Backup Name ein. Es ist empfehlenswert, dass aus dem Namen der gesicherte Cluster und das Datum sowie die Uhrzeit der Sicherung ersichtlich ist.

Für die Benennung von Clustern gelten die folgenden Einschränkungen:

- Er muss 1-40 alphanumerische Zeichen oder Bindestriche enthalten.
- Er muss mit einem Buchstaben beginnen.

- Er darf keine zwei aufeinanderfolgenden Bindestriche enthalten.
- Er darf nicht mit einem Bindestrich enden.

6. Wählen Sie Create Bucket.

Der Status des Clusters ändert sich in snapshotting.

Erstellen einer manuellen Sicherung (AWS CLI)

Manuelles Backup eines serverlosen Caches mit dem AWS CLI

Um eine manuelle Sicherung eines Caches mit dem zu erstellen AWS CLI, verwenden Sie den `create-serverless-snapshot` AWS CLI Vorgang mit den folgenden Parametern:

- `--serverless-cache-name` – der Name des Serverless-Caches, für den Sie ein Backup erstellen.
- `--serverless-cache-snapshot-name` – der Name des zu erstellenden Snapshots.

Für Linux, macOS oder Unix:

- ```
aws elasticache create-serverless-snapshot \
 --serverless-cache-name CacheName \
 --serverless-cache-snapshot-name bkup-20231127
```

Für Windows:

- ```
aws elasticache create-serverless-snapshot ^  
    --serverless-cache-name CacheName ^  
    --serverless-cache-snapshot-name bkup-20231127
```

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 letztes Backup erstellen, wenn Sie einen serverlosen Memcached-Cache 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 der AWS Management Console](#)

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 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 [delete-serverless-cache](#) 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

Kopieren eines Backups

Sie können eine Kopie jedes beliebigen Backups erstellen. Dabei spielt es keine Rolle, ob es automatisch oder manuell erstellt wurde.

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.

Wiederherstellen aus einem Backup in einen neuen Cache

Sie können ein vorhandenes Backup in einem neuen serverlosen Cache Cluster wiederherstellen.

Wiederherstellen eines Backups in einem Serverless-Cache (Konsole)

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 Serverless-Cache (AWS CLI)

So stellen Sie ein Backup in einem neuen Serverless-Cache wieder her (AWS CLI)

Im folgenden AWS CLI Beispiel wird ein neuer Cache mit Daten aus einem Backup erstellt `create-serverless-cache` und daraus importiert.

Für Linux, macOS oder Unix:

Für Windows:

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached \  
  --snapshot-arns-to-restore Snapshot-ARN
```

Für Windows:

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



```
--snapshot-arns-to-restore Snapshot-ARN
```

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, der AWS CLI, 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 AWS Management Console an 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.

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](#).

Engine-Versionen und Upgrades

Dieser Abschnitt beschreibt die unterstützten Memcached-Engine-Versionen, und wie ein Upgrade dafür ausgeführt wird.

Themen

- [Unterstützte ElastiCache \(Memcached\) Versionen](#)
- [Engine-Versionen und Upgrades](#)
- [So führen Sie ein Upgrade von Engine-Versionen aus](#)

Unterstützte ElastiCache (Memcached) Versionen

ElastiCache unterstützt die folgenden Memcached-Versionen und aktualisiert auf neuere Versionen. Achten Sie beim Upgrade auf eine neuere Version sorgfältig auf die Bedingungen, die Ihr Upgrade fehlschlagen lassen, wenn sie nicht erfüllt werden.

ElastiCache für Memcached-Versionen

- [Memcached-Version 1.6.17](#)
- [Memcached-Version 1.6.17](#)
- [Memcached-Version 1.6.12](#)
- [Memcached-Version 1.6.6](#)
- [Memcached-Version 1.5.16](#)
- [Memcached-Version 1.5.10](#)
- [Memcached-Version 1.4.34](#)
- [Memcached-Version 1.4.33](#)
- [Memcached-Version 1.4.24](#)
- [Memcached-Version 1.4.14](#)
- [Memcached-Version 1.4.5](#)

Memcached-Version 1.6.17

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.22 hinzu. Es sind keine neuen Funktionen enthalten, aber Fehlerbehebungen und kumulative Updates von [Memcached 1.6.18](#).

[Weitere Informationen finden Sie unter 1622 bei Memcached am. ReleaseNotes](#) GitHub

Memcached-Version 1.6.17

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.17 hinzu. Es sind keine neuen Funktionen enthalten, aber Fehlerbehebungen und kumulative Updates von [Memcached 1.6.17](#).

[Weitere Informationen finden Sie unter 1617 bei Memcached am. ReleaseNotes](#) GitHub

Memcached-Version 1.6.12

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.12 und Verschlüsselung bei der Übertragung hinzu. Es enthält Fehlerkorrekturen und kumulative Aktualisierungen von [Memcached 1.5.16](#).

Weitere Informationen finden Sie unter [1612 bei Memcached am. ReleaseNotes](#) GitHub

Memcached-Version 1.6.6

ElastiCache (Memcached) fügt Unterstützung für Memcached Version 1.6.6 hinzu. [Sie enthält keine neuen Funktionen, aber Bugfixes und kumulative Updates von Memcached 1.5.16](#). ElastiCache (Memcached) bietet keine Unterstützung für Extstore.

Weitere Informationen finden Sie in Memcached unter [ReleaseNotes166](#). GitHub

Memcached-Version 1.5.16

ElastiCache for Memcached fügt Unterstützung für Memcached Version 1.5.16 hinzu. Es sind keine neuen Funktionen enthalten, dafür aber Fehlerbehebungen und kumulative Updates von [Memcached 1.5.14](#) und [Memcached 1.5.15](#).

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.5.16 auf [Memcached](#) am. GitHub

Memcached-Version 1.5.10

ElastiCache für Memcached unterstützt Version 1.5.10 die folgenden Memcached-Funktionen:

- Automatisierter Slab-Ausgleich
- Schnelleres Suchen von Hash-Tabellen durch murmur3-Algorithmus
- Segmentierter LRU-Algorithmus
- LRU-Crawler, um im Hintergrund Speicher zurückzugewinnen
- `--enable-seccomp`: eine Kompilierungszeitoption

Es werden auch die Parameter `no_modern` und `inline_ascii_resp` eingeführt. Weitere Informationen finden Sie unter [Parameteränderungen für Memcached 1.5.10](#).

Seit ElastiCache Memcached Version 1.4.34 wurden unter anderem folgende Verbesserungen hinzugefügt:

- Kumulative Korrekturen wie ASCII-Multiget, CVE-2017-9951 und Limit-Crawler for metadumper
- Verbesserte Verbindungsverwaltung durch das Schließen von Verbindungen beim Erreichen der maximalen Verbindungsanzahl
- Verbesserte Verwaltung der Elementgröße bei Elementen, deren Größe 1 MB überschreitet
- Bessere Leistung und Optimierung des Speichermehraufwands durch Reduzierung der Speicheranforderungen pro Element um wenige Bytes

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.5.10 auf [Memcached](#) am. GitHub

Memcached-Version 1.4.34

ElastiCache für Memcached fügt Version 1.4.34 keine neuen Funktionen zu Version 1.4.33 hinzu. Version 1.4.34 dient der Fehlerbehebung und ist größer als für eine solche Version üblich.

Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.4.34 auf [Memcached](#) am. GitHub

Memcached-Version 1.4.33

Es wurden folgende Memcached-Verbesserungen seit Version 1.4.24 eingeführt:

- Die Fähigkeit, alle Metadaten für eine bestimmte Slab-Klasse, eine Liste der Slab-Klassen oder alle Slab-Klassen zu sichern. Weitere Informationen finden Sie unter [Memcached 1.4.31 Release Notes](#).
- Verbesserter Support für große Elemente über den Standardwert von 1 MB. Weitere Informationen finden Sie unter [Memcached 1.4.29 Release Notes](#).
- Die Fähigkeit, anzugeben, wie lange sich ein Client im Leerlauf befinden kann, bevor er zum Schließen aufgefordert wird.

Die Fähigkeit, den für Memcached verfügbaren Speicherplatz dynamisch zu erhöhen, ohne den Cluster neu starten zu müssen. Weitere Informationen finden Sie unter [Memcached 1.4.27 Release Notes](#).

- Die Protokollierung von `fetchers`, `mutations` und `evictions` wird jetzt unterstützt. Weitere Informationen finden Sie unter [Memcached 1.4.26 Release Notes](#).
- Freier Speicher kann für einen globalen Pool wieder freigegeben und neuen Slab-Klassen zugewiesen werden. Weitere Informationen finden Sie unter [Memcached 1.4.25 Release Notes](#).

- Mehrere Fehlerbehebungen
- Einige neue Befehle und Parameter Eine Liste finden Sie hier: [In Memcached 1.4.33 hinzugefügte Parameter](#).

Memcached-Version 1.4.24

Es wurden folgende Memcached-Verbesserungen seit Version 1.4.14 eingeführt:

- Die LRU-Verwaltung (Least Recently Used) mit einem Hintergrundprozess wurde verbessert.
- Die Option, jenkins oder murmur3 als Hash-Algorithmus zu verwenden, wurde hinzugefügt.
- Einige neue Befehle und Parameter Eine Liste finden Sie hier: [In Memcached 1.4.24 hinzugefügte Parameter](#).
- Mehrere Fehlerbehebungen

Memcached-Version 1.4.14

Es wurden folgende Memcached-Verbesserungen seit Version 1.4.5 eingeführt:

- Verbessertes Slab-Ausgleich.
- Performance und Skalierbarkeit wurden verbessert.
- Der Befehl touch wurde eingeführt, um das Ablaufdatum eines vorhandenen Elements zu aktualisieren, ohne es abzurufen.
- Automatische Erkennung – die Möglichkeit für Client-Programme, automatisch alle Cache-Knoten in einem Cluster zu ermitteln und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten.

Memcached-Version 1.4.5

Memcached Version 1.4.5 war die erste Engine und Version, die von Amazon unterstützt wurde ElastiCache (Memcached).

Engine-Versionen und Upgrades

MAJOR Versionen sind für API inkompatible Änderungen und MINOR Versionen für neue Funktionen, die abwärtskompatibel hinzugefügt wurden. PATCH Versionen sind für abwärtskompatible Bugfixes und nicht funktionale Änderungen vorgesehen.

Note

Wenn ein OSS Redis-Cluster in einer oder mehreren Regionen repliziert wird, wird die Engine-Version für sekundäre Regionen und dann für die primäre Region aktualisiert.

Versionsverwaltung für Serverless ElastiCache

ElastiCache Serverless wendet automatisch die neueste Version MINOR und PATCH Softwareversion auf Ihren Cache an, ohne dass dies Auswirkungen oder Ausfallzeiten auf Ihre Anwendung hat. Von Ihrer Seite aus ist keine Aktion erforderlich.

Wenn eine neue MAJOR Version verfügbar ist, sendet Ihnen ElastiCache Serverless eine Benachrichtigung in der Konsole und ein Ereignis in EventBridge. Sie können wählen, ob Sie Ihren Cache auf die neueste Hauptversion aktualisieren möchten, indem Sie Ihren Cache mithilfe der Konsole ändernCLI, oder API und die neueste Engine-Version auswählen.

Versionsverwaltung für selbst entworfene Cluster ElastiCache

Wenn Sie mit selbst entworfenen ElastiCache Clustern arbeiten, können Sie steuern, wann die Software, die Ihren Cache-Cluster unterstützt, auf neue Versionen aktualisiert wird, die von ElastiCache unterstützt werden. Sie können steuern, wann Ihr Cache auf die neuesten verfügbaren MAJOR Versionen und PATCH aktualisiert werden soll. MINOR 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, die Ihren Cache-Cluster unterstützt, 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.

Um auf eine neue Memcached-Version upzugraden, ändern Sie Ihren Cache-Cluster und geben Sie die neue Engine-Version an, die Sie verwenden möchten. Das Upgrade auf eine neuere Memcached-Version ist ein destruktiver Prozess – Sie verlieren Ihre Daten und beginnen mit einem kalten Cache. Weitere Informationen finden Sie unter [Ändern von Clustern](#).

Beim Upgraden einer älteren Memcached-Version auf Memcached Version 1.4.33 oder höher sind folgende Anforderungen zu beachten. `CreateCacheCluster` und `ModifyCacheCluster` schlagen unter den folgenden Bedingungen fehl:

- Wenn `slab_chunk_max > max_item_size`.
- Wenn `max_item_size modulo slab_chunk_max != 0`.
- Wenn `max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)`.

Der Wert `(max_cache_memory - memcached_connections_overhead)` ist der für Daten nutzbare Speicher des Knotens. Weitere Informationen finden Sie unter [Overhead von Memcached-Verbindungen](#).

Überlegungen zum Upgrade bei der Arbeit mit selbst entworfenen Clustern

Note

Die folgenden Überlegungen gelten nur für Upgrades von selbst entworfenen Clustern. Sie gelten nicht für Serverless. ElastiCache

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 für den unwahrscheinlichen Fall einer kritischen Sicherheitslücke im System oder in der Cache-Software zu patchen.
- Da die Memcached-Engine keine Persistenz unterstützt, stellen Versions-Upgrades der Memcached-Engine immer einen Störfall dar, bei dem alle Cache-Daten im Cluster gelöscht werden.

So führen Sie ein Upgrade von Engine-Versionen aus

Um für Ihr Cluster Versions-Upgrades zu starten, ändern sie es und geben eine neuere Engine-Version an. Sie können dies mit der ElastiCache Konsole, dem oder dem AWS CLI tun: ElastiCache API

- Informationen zur AWS Management Console Verwendung von finden Sie unter — [Cluster über die Konsole ändern](#).
- Informationen zur Verwendung von finden Sie unter [Modifizieren von Clustern mit dem CLI](#). AWS CLI
- Informationen zur Verwendung von finden Sie unter [Modifizieren von Clustern mit dem API](#). ElastiCache API

So führen Sie ein Upgrade von Engine-Versionen aus

Um für Ihr Cluster Versions-Upgrades zu starten, ändern sie es und geben eine neuere Engine-Version an. Sie können dies mithilfe der ElastiCache Konsole AWS CLI, der oder der ElastiCache API tun:

- Informationen zur Verwendung von AWS Management Console finden Sie unter —[Mit dem AWS Management Console](#).
- Informationen zur Verwendung von AWS CLI finden Sie unter [Verwenden Sie den AWS CLI](#).
- Informationen zur Verwendung der ElastiCache API finden Sie unter [Verwenden der ElastiCache API](#).

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

- [Bewährte Methoden für Memcached-Clients](#)
- [Unterstützte Memcached-Befehle](#)
- [Caching-Strategien](#)

Bewährte Methoden für Memcached-Clients

Weitere Informationen zu bewährten Methoden für die Interaktion mit ElastiCache Ressourcen mit häufig verwendeten Open-Source-Clientbibliotheken von Memcached finden Sie in den folgenden Themen.

Themen

- [Konfigurieren Ihres ElastiCache-Clients für ein effizientes Load Balancing](#)
- [Beispiele für IPv6-Clients](#)

Konfigurieren Ihres ElastiCache-Clients für ein effizientes Load Balancing

Note

Dieser Abschnitt trifft auf selbst entworfene Memcached-Cluster mit mehreren Knoten zu.

Um mehrere ElastiCache Memcached-Knoten effektiv zu verwenden, müssen Sie Ihre Cache-Schlüssel auf die Knoten verteilen können. Eine einfache Methode für den Lastausgleich eines Clusters mit n Knoten besteht in der Berechnung des Hash-Wertes des Objektschlüssels und Verschiebung des Ergebnisses um $n - \text{hash}(\text{key}) \bmod n$. Der resultierende Wert (0 bis $n-1$) ist die Nummer des Knotens, auf dem Sie das Objekt platzieren.

Diese Methode ist einfach und funktioniert gut, solange die Anzahl der Knoten (n) konstant bleibt. Wenn Sie einen Knoten zum Cluster hinzufügen oder daraus entfernen, beträgt die Anzahl der zu verschiebenden Schlüssel jedoch jedes Mal $(n - 1) / n$ (dabei entspricht n der neuen Anzahl von Knoten). Diese Methode führt daher dazu, dass eine große Anzahl von Schlüsseln verschoben werden. Dies ist mit einer großen Anzahl anfänglicher Cache-Fehlelemente verbunden, insbesondere bei einer umfangreichen Anzahl von Knoten. Die Skalierung von 1 auf 2 Knoten führt bestenfalls dazu, dass $(2-1) / 2$ (50 %) der Schlüssel verschoben werden. Eine Skalierung von 9 auf 10 Knoten hat bestenfalls zur Folge, dass $(10-1) / 10$ (90 %) der Schlüssel verschoben werden. Ist die Aufwärtsskalierung durch einen sprunghaften Anstieg im Datenverkehr bedingt, ist eine große Anzahl von Cache-Fehlelementen nicht wünschenswert. Eine große Anzahl von Cache-Fehlelementen führt zu Anfragen bei der Datenbank, die aufgrund des sprunghaften Anstiegs im Datenverkehr bereits überlastet ist.

Die Lösung zu diesem Dilemma ist konsistentes Hashing. Beim konsistenten Hashing wird ein Algorithmus verwendet, bei dem jedes Mal, wenn ein Knoten zu einem Cluster hinzugefügt oder daraus entfernt wird, die Anzahl der zu verschiebenden Schlüssel ungefähr $1 / n$ beträgt (wobei n die neue Anzahl von Knoten ist). Eine Skalierung von 1 auf 2 Knoten hat schlimmstenfalls zur Folge, dass $1/2$ (50 Prozent) der Schlüssel verschoben werden. Eine Skalierung von 9 auf 10 Knoten hat zur Folge, dass $1/10$ (10 Prozent) der Schlüssel verschoben werden.

Sie als Benutzer steuern, welcher Hashalgorithmus für Mehrknoten-Cluster verwendet wird. Wir empfehlen, dass Sie Ihre Clients zur Verwendung von konsistentem Hashing konfigurieren. Glücklicherweise sind viele Memcached-Clientbibliotheken in den meisten gängigen Sprachen verfügbar, die konsistentes Hashing implementieren. Überprüfen Sie die Dokumentation der

von Ihnen verwendeten Bibliothek darauf, ob sie konsistentes Hashing unterstützt und wie es implementiert wird.

Wenn Sie in Java, PHP oder .NET arbeiten, empfehlen wir Ihnen, eine der Amazon-ElastiCache-Clientbibliotheken zu verwenden.

Konsistentes Hashing mithilfe von Java

Der ElastiCache Memcached Java-Client basiert auf dem Open-Source-Spymemcached-Java-Client, der über konsistente Hashing-Funktionen verfügt. Die Bibliothek enthält eine `KetamaConnectionFactory`-Klasse, mit der konsistentes Hashing implementiert wird. Konsistentes Hashing ist in `spymemcached` standardmäßig deaktiviert.

Weitere Informationen finden Sie in der `KetamaConnectionFactory`-Dokumentation unter [KetamaConnectionFactory](#).

Konsistentes Hashing mit PHP

Der ElastiCache Memcached PHP-Client ist ein Wrapper um die integrierte Memcached-PHP-Bibliothek. Konsistentes Hashing wird von der Memcached-PHP-Bibliothek standardmäßig deaktiviert.

Verwenden Sie den folgenden Code, um konsistentes Hashing zu aktivieren.

```
$m = new Memcached();  
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

Zusätzlich zum obigen Code wird empfohlen, ebenfalls `memcached.sess_consistent_hash` in der Datei `php.ini` einzuschalten.

Weitere Informationen erhalten Sie in der Dokumentation zur Laufzeitkonfiguration für Memcached-PHP unter <http://php.net/manual/en/memcached.configuration.php>. Beachten Sie insbesondere den Parameter `memcached.sess_consistent_hash`.

Konsistentes Hashing mit .NET

Der ElastiCache Memcached .NET-Client ist ein Wrapper um `Enyim Memcached`. Konsistentes Hashing wird vom `Enyim-Memcached-Client` standardmäßig aktiviert.

Weitere Informationen erhalten Sie in der Dokumentation zu `memcached/locator` unter <https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator>.

Beispiele für IPv6-Clients

Note

Dieser Abschnitt trifft auf selbst entworfene Memcached-Cluster zu.

ElastiCache ist mit Open-Source-Memcached kompatibel. Das bedeutet, dass Open-Source-Clients für Memcached, die IPv6-Verbindungen unterstützen, in der Lage sein sollten, eine Verbindung zu IPv6-fähigen (Memcached) Clustern herzustellen. ElastiCache Darüber hinaus wurden die folgenden Clients speziell für die Verwendung mit allen unterstützten Netzwerktopologien validiert:

Im Folgenden finden Sie bewährte Methoden für die Interaktion mit IPv6-fähigen ElastiCache Ressourcen mit häufig verwendeten 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

Validierte Clients:

- [AWS ElastiCache Cluster-Client Memcached für PHP — Version *3.6.2](#)
- [AWS ElastiCache Cluster Client Memcached für Java](#) — Neuester Master auf Github

Konfigurieren eines bevorzugten Protokolls für Dual-Stack-Cluster

Bei Memcached-Clustern können Sie mit dem IP-Discovery-Parameter das Protokoll steuern, das Clients für die Verbindung mit den Knoten im Cluster verwenden. Der IP-Discovery-Parameter kann entweder auf IPv4 oder IPv6 festgelegt werden.

Der IP-Erkennungsparameter steuert das IP-Protokoll, das in der Cluster-Ausgabe der Konfiguration verwendet wird. Dies wiederum bestimmt das IP-Protokoll, das von Clients verwendet wird, die Autodiscovery für ElastiCache (Memcached-) Cluster unterstützen.

Eine Änderung vom IP-Discovery führt zu keinen Ausfallzeiten für verbundene Clients. Die Weiterleitung der Änderungen wird jedoch einige Zeit dauern.

Überwachen Sie die Ausgabe von `getAvailableNodeEndpoints` für Java und für Php überwachen Sie die Ausgabe von `getServerList`. Sobald die Ausgabe dieser Funktionen IPs für

alle Knoten im Cluster meldet, die das aktualisierte Protokoll verwenden, wurden die Änderungen vollständig übertragen.

Java-Beispiel:

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

do {
    nodes =
    client.getAvailableNodeEndpoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors.toList());

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByName(node));
    } catch (UnknownHostException ignored) {}
    return false;
})));
```

Php-Beispiel:

```
$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
    # The PHP memcached client only updates the server list if the polling interval has
    expired and a
    # command is sent
    $client->get('test');

    $nodes = $client->getServerList();

    sleep(1);
    $target_ips_count = 0;
```

```
// For IPv4 use FILTER_FLAG_IPV4
$target_ips_count = count(array_filter($nodes, function($node) { return
filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));

} while (count($nodes) !== $target_ips_count);
```

Alle vorhandenen Client-Verbindungen, die vor der Aktualisierung von IP Discovery erstellt wurden, werden weiterhin mit dem alten Protokoll verbunden. Alle validierten Clients stellen mithilfe des neuen IP-Protokolls automatisch wieder eine Verbindung mit dem Cluster her, sobald die Änderungen in der Ausgabe der Cluster-Erkennungsbefehle erkannt wurden. Dies hängt jedoch von der Implementierung des Clients ab.

TLS-fähige Dual-Stack-Cluster ElastiCache

Wenn TLS für ElastiCache Cluster aktiviert ist, geben die Cluster-Erkennungsfunktionen Hostnamen statt IPs `config get cluster` 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.

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
```

Unterstützte Memcached-Befehle

ElastiCache Serverless for Memcached unterstützt alle [Memcached-Befehle](#) in Open Source Memcached 1.6 mit Ausnahme der folgenden:

- Client-Verbindungen erfordern TLS, weshalb das UDP-Protokoll nicht unterstützt wird.
- Das Binärprotokoll wird nicht unterstützt, da es in Memcached 1.6 offiziell [veraltet](#) ist.
- GET/GETS-Befehle sind auf 16 KB begrenzt, um einen möglichen DoS-Angriff auf den Server durch Abrufen einer großen Anzahl von Schlüsseln zu vermeiden.
- Ein verzögerter `flush_all`-Befehl wird mit `CLIENT_ERROR` zurückgewiesen.
- Befehle, die die Engine konfigurieren oder interne Informationen zum Engine-Status oder zu Protokollen preisgeben, werden nicht unterstützt. Dazu gehören:
 - Für STATS-Befehle werden nur `stats` und `stats reset` unterstützt. Für andere Varianten wird `ERROR` zurückgegeben.
 - `lru / lru_crawler` – Änderung der LRU- und LRU-Crawler-Einstellungen
 - `watch` – beobachtet Memcached-Serverprotokolle
 - `verbosity` – konfiguriert die Protokollstufe des Servers
 - `me`- Der Befehl `meta debug (me)` wird nicht unterstützt

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 speicherinterner Schlüsselwertspeicher, der sich zwischen Ihrer Anwendung und dem Datenspeicher (Datenbank) befindet, auf den sie zugreift. Immer wenn Ihre Anwendung Daten anfordert, sendet sie die Anfrage zuerst an den Cache. ElastiCache Wenn die Daten im Cache vorhanden und aktuell sind, werden die Daten an Ihre Anwendung ElastiCache zurückgegeben. Wenn die Daten nicht im Cache vorhanden sind oder abgelaufen sind, fordert Ihre Anwendung die Daten von Ihrem Datenspeicher an. Ihr Datenspeicher gibt die Daten dann an Ihre Anwendung zurück. Als nächstes schreibt Ihre Anwendung die vom Speicher empfangenen Daten in den Cache. Auf diese Weise können sie bei der nächsten Anforderung schneller abgerufen werden.

Ein Cache-Treffer tritt auf, wenn sich Daten im Cache befinden und nicht abgelaufen sind:

1. Die Anwendung fordert Daten aus dem Cache an.
2. Der Cache gibt die Daten an die Anwendung zurück.

Ein Cache-Fehlertreffer tritt auf, wenn sich Daten nicht im Cache befinden oder abgelaufen sind:

1. Die Anwendung fordert Daten vom Cache an.
2. Der Cache enthält die angeforderten Daten nicht und gibt daher `null` zurück.
3. Ihre Anwendung fordert die Daten an und erhält sie aus der Datenbank.
4. Ihre Anwendung aktualisiert den Cache mit den neuen Daten.

Vor- und Nachteile von Lazy Loading

Die Vorteile von Lazy Loading sind:

- Es werden nur die angeforderten Daten im Cache abgelegt.

Da die meisten Daten nie angefordert werden, wird durch Lazy Loading vermieden, dass der Cache mit nicht angeforderten Daten gefüllt wird.

- Knotenfehler sind für Ihre Anwendung nicht fatal.

Wenn ein Knoten ausfällt und durch einen neuen, leeren Knoten ersetzt wird, funktioniert Ihre Anwendung weiterhin, allerdings mit erhöhter Latenz. Wenn Anforderungen an den neuen Knoten gestellt werden, führt jeder Cache-Fehlertreffer zu einer Abfrage der Datenbank. Gleichzeitig wird die Datenkopie dem Cache hinzugefügt, so dass nachfolgende Anforderungen aus dem Cache abgerufen werden.

Die Nachteile von Lazy Loading sind:

- Bei Cache-Fehlschlägen gibt es Verzögerungen. Jeder Cache-Fehlertreffer führt zu drei Übertragungsvorgängen:

1. Anfängliche Anforderung von Daten aus dem Cache
2. Abfrage der Daten aus der Datenbank
3. Schreiben der Daten in den Cache

Diese Fehltreffer 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. Memcached gibt diesen Wert in Sekunden an. Wenn eine Anwendung versucht, einen abgelaufenen Schlüssel zu lesen, gilt der Schlüssel als nicht gefunden. Die Datenbank wird nach dem Schlüssel abgefragt und der Cache aktualisiert. Dieser Ansatz garantiert nicht, dass ein Wert nicht veraltet ist. Er verhindert jedoch, dass die Daten zu veraltet werden, und erfordert, dass Werte im Cache gelegentlich aus der Datenbank aktualisiert werden.

[Weitere Informationen finden Sie im OSS-Befehl Memcached Command. `setset`](#)

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 \(Memcached\)](#)

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

- [Verwaltung der Wartung](#)
- [Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen](#)

Verwaltung der Wartung

Jeder Cluster verfügt über ein wöchentliches Wartungsfenster, während dem alle Systemänderungen angewendet werden. Wenn Sie bei der Erstellung oder Änderung eines Clusters kein bevorzugtes Wartungsfenster angeben, ElastiCache weisen Sie innerhalb des Wartungsfensters Ihrer Region an einem zufällig ausgewählten Wochentag ein 60-minütiges Wartungsfenster zu.

Das 60-minütige Wartungsfenster wird zufällig aus einem 8-Stunden-Zeitraum pro Region ausgewählt. Die folgende Tabelle listet die Blöcke für jede Region auf, von denen die Standard-Wartungsfenster zugewiesen werden. Sie können ein bevorzugtes Wartungsfenster außerhalb des Wartungsfensterblocks der Region auswählen.

Regionscode	Name der Region	Regionale Wartungsfenster
ap-northeast-1	Region Asien-Pazifik (Tokio)	13:00 - 21:00 UHR UTC
ap-northeast-2	Region Asien-Pazifik (Seoul)	12:00 - 20:00 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

Regionscode	Name der Region	Regionale Wartungsfenster
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
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	AWS GovCloud (US) Region	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 einen Cluster

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 ändern und einen Zeitraum mit einer Dauer von bis zu 24 Stunden festlegen, in dem alle angeforderten Wartungsaktivitäten durchgeführt werden sollen. In diesem Zeitraum werden alle verzögerten oder ausstehenden Cluster-Änderungen, die Sie angefordert haben, ausgeführt.

Note

Wenn Sie Änderungen am Knotentyp und/oder Engine-Upgrades sofort anwenden möchten, AWS Management Console wählen Sie das Feld Jetzt anwenden aus. Andernfalls werden diese Änderungen während des nächsten geplanten Wartungsfensters angewendet. Informationen zur Verwendung der API finden Sie unter [modify-replication-group](#) oder [modify-cache-cluster](#).

Weitere Informationen

Informationen zu Ihrem Wartungsfenster und dem Austausch von Knoten finden Sie unter:

- [ElastiCache Wartung](#) — Häufig gestellte Fragen zur Wartung und zum Austausch von Knoten
- [Ersetzen von Knoten](#) – Verwalten des Knotenaustauschs
- [Einen ElastiCache Cluster ändern](#) – Ändern des Wartungsfensters für einen Cluster

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 [Memcached-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 des Memcached-Knotentyps](#).

Note

Eine Liste der Memcached-spezifischen Parameter finden Sie unter [Memcached-spezifische Parameter](#).

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](#)

Parameterverwaltung

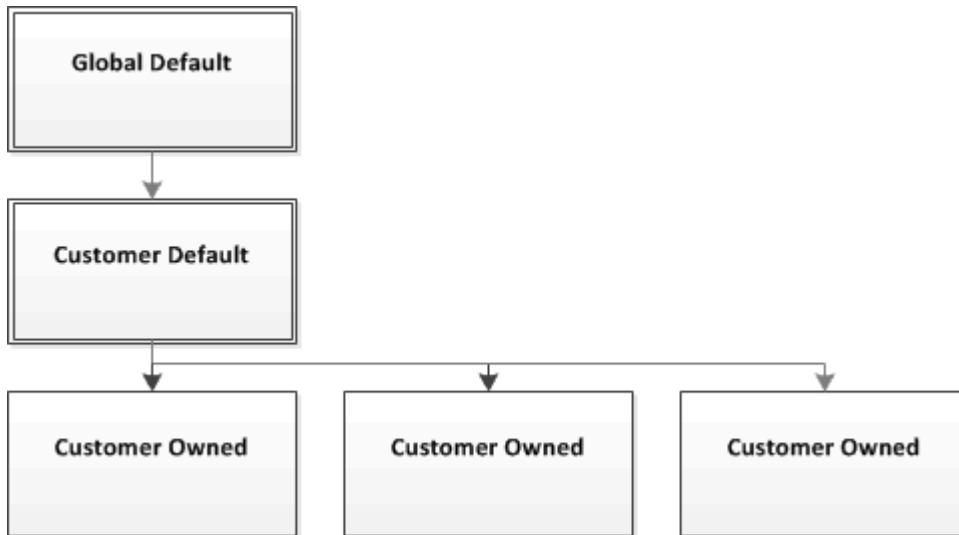
Parameter werden zur leichteren Parameterverwaltung in benannte Gruppen zusammengruppiert. Eine Parametergruppe stellt eine Kombination spezifischer Werte für die Parameter dar, die der Engine-Software während des Systemstarts übergeben werden. Diese Werte bestimmen, wie sich die Engine-Prozesse auf 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. Memcached Version 1.4.8 ausführt, können Sie nur Parametergruppen, Standard oder benutzerdefiniert, von der Memcached 1.4-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 sofort auf den Cluster angewendet. Dies gilt unabhängig davon, ob Sie die Parametergruppe des Clusters oder einen Parameterwert innerhalb der Parametergruppe des Clusters ändern. Anhand der Spalte Changes Take Effect (Änderungen treten in Kraft) in den Tabellen für [Memcached-spezifische Parameter](#) können Sie bestimmen, wann eine bestimmte Parameteränderung übernommen wird. Weitere Informationen zum Neustarten der Knoten eines Clusters finden Sie unter [Neustarten von Clustern](#).

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 [Memcached-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. memcached1.4 , 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 memcached1.4-Familie als Vorlage eine Parametergruppe namens myMem14 erstellt.

Für Linux, macOS oder Unix:

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --cache-parameter-group-family memcached1.4 \  
  --description "My first parameter group"
```

Für Windows:

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --cache-parameter-group-family memcached1.4 ^  
  --description "My first parameter group"
```

Die Ausgabe dieses Befehls sollte in etwa folgendermaßen aussehen.

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myMem14",  
    "CacheParameterGroupFamily": "memcached1.4",  
    "Description": "My first parameter group"  
  }  
}
```

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 `memcached1.4`.
- `Description` Eine vom Benutzer eingegebene Beschreibung der Parametergruppe.

Example

Im folgenden Beispiel wird anhand der `memcached1.4`-Familie als Vorlage eine Parametergruppe namens `myMem14` erstellt.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheParameterGroup  
&CacheParameterGroupFamily=memcached1.4  
&CacheParameterGroupName=myMem14  
&Description=My%20first%20parameter%20group  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sollte in etwa folgendermaßen aussehen.

```
<CreateCacheParameterGroupResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <CreateCacheParameterGroupResult>  
    <CacheParameterGroup>  
      <CacheParameterGroupName>myMem14</CacheParameterGroupName>  
      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>  
      <Description>My first parameter group</Description>  
    </CacheParameterGroup>  
  </CreateCacheParameterGroupResult>  
</ResponseMetadata>
```

```
<RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>  
</ResponseMetadata>  
</CreateCacheParameterGroupResponse>
```

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 AWS CLI, der oder der ElastiCache API auflisten.

Auflisten von Parametergruppen nach Namen (Konsole)

Das folgende Verfahren zeigt, wie Sie mit der ElastiCache Konsole eine Liste der Parametergruppen anzeigen können.

Um Parametergruppen mithilfe der ElastiCache Konsole aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.

Auflisten von Parametergruppen nach Namen (AWS CLI)

Verwenden Sie den Befehl, um mit dem AWS CLI eine Liste von Parametergruppen zu generieren `describe-cache-parameter-groups`. Wenn Sie den Namen einer Parametergruppe angeben, wird nur die betreffende Parametergruppe aufgelistet. Wenn Sie keinen Namen einer Parametergruppe angeben, werden bis zu `--max-records` Parametergruppen aufgelistet. In beiden Fälle werden Name, Familie und Beschreibung der Parametergruppe aufgelistet.

Example

Der folgenden Beispielcode listet die Parametergruppe `myMem14` auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name myMem14
```

Für Windows:

```
aws elasticache describe-cache-parameter-groups ^  
  --cache-parameter-group-name myMem14
```

Die Ausgabe dieses Befehls sieht folgendermaßen aus und listet Name, Familie und Beschreibung der Parametergruppe auf.

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "myMem14",
      "CacheParameterGroupFamily": "memcached1.4",
      "Description": "My first parameter group"
    }
  ]
}
```

Example

Der 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](#).

Parametergruppen nach Namen auflisten (API) ElastiCache

Verwenden Sie die DescribeCacheParameterGroups Aktion, um mithilfe der ElastiCache API eine Liste von Parametergruppen zu generieren. Wenn Sie den Namen einer Parametergruppe angeben, wird nur die betreffende Parametergruppe aufgelistet. Wenn Sie keinen Namen einer Parametergruppe angeben, werden bis zu MaxRecords Parametergruppen aufgelistet. In beiden Fälle werden Name, Familie und Beschreibung der Parametergruppe aufgelistet.

Example

Der folgenden Beispielcode listet die Parametergruppe myMem14 auf.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&CacheParameterGroupName=myMem14
&SignatureVersion=4
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sieht folgendermaßen aus und listet Name, Familie und Beschreibung jeder Parametergruppe auf.

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>myMem14</CacheParameterGroupName>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
        <Description>My custom Memcached 1.4 parameter group</Description>
      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Example

Der folgenden Beispielcode listet bis zu 10 Parametergruppen auf.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheParameterGroups
&MaxRecords=10
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Die Antwort 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>
```

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 `myMem14` auf.

Für Linux, macOS oder Unix:

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myMem14
```

Für Windows:

```
aws elasticache describe-cache-parameters ^  
  --cache-parameter-group-name myMem14
```

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 myMem14 auf.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheParameters  
&CacheParameterGroupName=myMem14  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

Die Antwort auf diese Aktion sieht in etwa folgendermaßen aus. Diese Antwort wurde abgeschnitten.

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParametersResult>  
    <CacheClusterClassSpecificParameters>  
      <CacheNodeTypeSpecificParameter>  
        <DataType>integer</DataType>  
        <Source>system</Source>  
        <IsModifiable>>false</IsModifiable>  
        <Description>The maximum configurable amount of memory to use to store items,  
in megabytes.</Description>  
        <CacheNodeTypeSpecificValues>  
          <CacheNodeTypeSpecificValue>  
            <Value>1000</Value>  
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>
```

```
<CacheNodeTypeSpecificValue>
  <Value>6000</Value>
  <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
</CacheNodeTypeSpecificValue>
<CacheNodeTypeSpecificValue>
  <Value>7100</Value>
  <CacheClusterClass>cache.m1.large</CacheClusterClass>
</CacheNodeTypeSpecificValue>
<CacheNodeTypeSpecificValue>
  <Value>1300</Value>
  <CacheClusterClass>cache.m1.small</CacheClusterClass>
</CacheNodeTypeSpecificValue>
```

...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 [Memcached-spezifische Parameter](#).

Ändern einer Parametergruppe (Konsole)

Das folgende Verfahren zeigt, wie Sie den Wert des `binding_protocol` Parameters mithilfe der ElastiCache Konsole ändern können. Dieses Verfahren ist bei der Bearbeitung des Wertes aller Parameter gleich.

So ändern Sie den Wert eines Parameters mithilfe der ElastiCache Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Eine Liste aller verfügbaren Parametergruppen finden Sie, wenn Sie im linken Navigationsbereich Parametergruppen auswählen.
3. Wählen Sie die Parametergruppe aus, die Sie bearbeiten möchten, indem Sie das Kontrollkästchen links neben dem Namen der Parametergruppe aktivieren.

Die Parameter der Parametergruppe werden unten auf dem Bildschirm aufgelistet. Sie können die Liste seitenweise durchlaufen, um alle Parameter zu sehen.

4. Um einen oder mehrere Parameter zu bearbeiten, wählen Sie Edit Parameters.
5. Scrollen Sie auf dem Bildschirm Edit Parameter Group: mit der linken und rechten Pfeiltaste, bis Sie zum Parameter `binding_protocol` gelangen. Geben Sie dann `ascii` in die Spalte Value ein.
6. Scrollen Sie auf dem Bildschirm Edit Parameter Group: mit der linken und rechten Pfeiltaste, bis Sie zum Parameter `cluster-enabled` gelangen. Geben Sie dann `yes` in die Spalte Value ein.
7. Wählen Sie Save Changes.
8. Den Namen des Parameters, den Sie in einem dieser Themen bearbeitet haben, finden Sie unter [Memcached-spezifische Parameter](#). Wenn als Zeitpunkt für die Übernahme der Parameteränderung After restart festgelegt ist, starten Sie jeden Cluster neu, der von dieser Parametergruppe Gebrauch macht. Weitere Informationen finden Sie unter [Neustarten von Clustern](#).

Ändern einer Parametergruppe (AWS CLI)

Um den Wert eines Parameters mit dem zu ändern AWS CLI, verwenden Sie den Befehl `modify-cache-parameter-group`.

Example

Den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, finden Sie unter [Memcached-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `chunk_size` und `chunk_size_growth_fact` auf der Parametergruppe `myMem14`.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --parameter-name-values \  
    ParameterName=chunk_size,ParameterValue=96 \  
    ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Für Windows:

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --parameter-name-values ^  
    ParameterName=chunk_size,ParameterValue=96 ^  
    ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

Die Ausgabe dieses Befehls sieht in etwa folgendermaßen aus.

```
{  
  "CacheParameterGroupName": "myMem14"  
}
```

Weitere Informationen finden Sie unter [modify-cache-parameter-group](#).

Ändern einer Parametergruppe (ElastiCache API)

Verwenden Sie die `ModifyCacheParameterGroup` Aktion, um die Parameterwerte einer Parametergruppe mithilfe der ElastiCache API zu ändern.

Example

Den Namen und die zulässigen Werte des Parameters, den Sie ändern möchten, finden Sie unter [Memcached-spezifische Parameter](#)

Der folgende Beispiel-Code legt den Wert zweier Parameter fest: `chunk_size` und `chunk_size_growth_fact` auf der Parametergruppe `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheParameterGroup  
&CacheParameterGroupName=myMem14  
&ParameterNameValues.member.1.ParameterName=chunk_size
```

```
&ParameterNameValues.member.1.ParameterValue=96
&ParameterNameValues.member.2.ParameterName=chunk_size_growth_fact
&ParameterNameValues.member.2.ParameterValue=1.5
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T192317Z
&Version=2015-02-02
&X-Amz-Credential=<credential>
```

Weitere Informationen finden Sie unter [ModifyCacheParameterGroup](#).

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 myMem14
```

Für Windows:

```
aws elasticache delete-cache-parameter-group ^  
  --cache-parameter-group-name myMem14
```

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 `myMem14`.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteCacheParameterGroup  
&CacheParameterGroupName=myMem14  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T192317Z  
&Version=2015-02-02  
&X-Amz-Credential=<credential>
```

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>fastex</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 data-bbox="1008 1087 1508 1451" 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
<code>slab_reassign</code>	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
<code>backlog_queue_limit</code>	Standard: 1024 Typ: Ganzzahl Veränderbar: Nein	Das Limit der Rückstandwarteschlange (Backlog Queue).
<code>binding_protocol</code>	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.

Important

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
dache.r6g.2xlarge	54084	8
dache.r6g.4xlarge	108347	16
dache.r6g.8xlarge	214577	32
dache.r6g.12xlarge	325400	48
dache.r6g.16xlarge	429154	64
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16

Knotentyp	max_cache_memory (in Megabyte)	num_threads
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

 Note

Alle T2-Instances werden in einer Amazon Virtual Private Cloud (Amazon VPC) erstellt.

Skalierung ElastiCache (Memcached)

Skalierung ElastiCache (Memcached)

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 die Cache-Datenspeicherung 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 beginnt das Löschen von Daten in Ihrem Cache mithilfe der LRU-Logik. Wenn Sie ein Maximum in ECPU/Sekunde festlegen und die Rechenauslastung Ihres Workloads diesen Wert überschreitet, ElastiCache beginnt die Drosselung von Memcached-Anfragen.

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 Vorkalibrierung 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 Vorkalibrierung

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 zwischen den Redis OSS-Steckplätzen und der Datenverkehr zwischen den Schlüsseln so einheitlich wie möglich sind.

Skalierungsgrenzen mithilfe der Konsole festlegen und AWS CLI

Skalierungsgrenzen mithilfe der AWS Konsole festlegen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie im Navigationsbereich die Engine, die in dem Cache ausgeführt wird, den Sie ändern möchten.
3. Es wird eine Liste der Caches der ausgewählten Engine angezeigt.
4. Wählen Sie den zu ändernden Cache aus, indem Sie das Optionsfeld links neben dem Cache-Namen anklicken.
5. Wählen Sie Aktionen und dann Ändern.
6. Legen Sie unter Nutzungslimits die entsprechenden Speicher - oder Rechenlimits fest.
7. Klicken Sie auf Änderungen in der Vorschau anzeigen und dann auf Änderungen speichern.

Einstellen von Skalierungsgrenzen mithilfe der AWS CLI

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 (Memcached) selbst entworfener Cluster

Die Datenmenge, die von einer Anwendung verarbeitet wird, ist selten statisch. Sie steigt und sinkt mit dem Unternehmenswachstum und unterliegt normalen Schwankungen im Bedarf. Wenn Sie Ihren Cache selbst verwalten, müssen Sie für Bedarfsspitzen ausreichend Hardware bereitstellen – dies kann kostspielig sein. Mit Amazon können ElastiCache Sie skalieren, um der aktuellen Nachfrage gerecht zu werden, und zahlen nur für das, was Sie tatsächlich nutzen. ElastiCache ermöglicht es Ihnen, Ihren Cache an die Nachfrage anzupassen.

Note

Wenn ein OSS Redis-Cluster über eine oder mehrere Regionen repliziert wird, werden diese Regionen der Reihe nach skaliert. Bei der Skalierung werden zuerst sekundäre Regionen und dann die primäre Region skaliert. Beim Herunterskalieren steht die primäre Region an erster Stelle und dann folgen alle sekundären Regionen.

Bei der Aktualisierung der Engine-Version lautet die Reihenfolge sekundäre Region und dann primäre Region.

Im Folgenden finden Sie das richtige Thema für die Skalierungsaktionen, die Sie ausführen möchten.

Skalierung von Memcached-Clustern

Aktion	Thema
Ausskalieren	Hinzufügen von Knoten zu einem Cluster
Einskalieren	Löschen von Knoten aus einem Cluster
Ändern von Knotentypen	Vertikale Skalierung von Memcached

Memcached-Cluster bestehen aus 1 bis 60 Knoten. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten zu bzw. aus dem Cluster nach oben bzw. nach unten skaliert.

[Wenn Sie mehr als 60 Knoten in einem Memcached-Cluster oder insgesamt mehr als 300 Knoten in einer AWS Region benötigen, füllen Sie das Antragsformular zur Erhöhung des ElastiCache Grenzwerts unter/aus. <https://aws.amazon.com/contact-us/elasticache-node-limit-request>](#)

Da Sie in einem Memcached-Cluster Ihre Daten auf alle vorhandenen Knoten verteilen können, müssen Sie selten auf einen Knotentyp mit mehr Speicher hochskalieren. Da Daten in der Memcached-Engine jedoch nicht beibehalten werden, müssen Sie, wenn Sie auf einen anderen Knotentyp skalieren, Ihren Cluster erstellen, der zunächst leer ist, bis er durch Ihre Anwendung befüllt wird.

Themen

- [Horizontale Skalierung von Memcached](#)
- [Vertikale Skalierung von Memcached](#)

Horizontale Skalierung von Memcached

Die Memcached-Engine unterstützt die Partitionierung Ihrer Daten über mehrere Knoten hinweg. Dadurch lassen sich Memcached-Cluster einfach horizontal skalieren. Ein Memcached-Cluster kann aus 1 bis 60 Knoten bestehen. Ein Memcached-Cluster wird einfach durch Hinzufügen oder Entfernen von Knoten horizontal skaliert.

[Wenn Sie mehr als 60 Knoten in einem Memcached-Cluster oder insgesamt mehr als 300 Knoten in einer AWS Region benötigen, füllen Sie das Formular zur Anforderung einer ElastiCache Limiterhöhung unter/aus. <https://aws.amazon.com/contact-us/elasticache-node-limit-request>](#)

In den folgenden Themen wird erläutert, wie Sie Memcached-Cluster durch Hinzufügen oder Entfernen von Knoten ein- oder ausskalieren.

- [Hinzufügen von Knoten zu einem Cluster](#)
- [Löschen von Knoten aus Ihrem Cluster](#)

Jedes Mal, wenn Sie die Anzahl der Knoten in Ihrem Memcached-Cluster ändern, müssen Sie zumindest einen Teil Ihres Keyspace neu zuordnen, sodass er dem richtigen Knoten zugeordnet ist. Detaillierte Informationen zum Lastausgleich Ihres Memcached-Clusters finden Sie unter [Konfigurieren Ihres ElastiCache-Clients für ein effizientes Load Balancing](#).

Wenn Sie für Ihr Memcached-Cluster Auto Discovery verwenden, müssen Sie die Endpunkte in Ihrer Anwendung nicht ändern, wenn Sie Knoten hinzufügen oder entfernen. Weitere Informationen zu Auto Discovery finden Sie unter [Automatische Erkennung von Knoten in Ihrem Cluster](#). Wenn Sie Auto Discovery nicht verwenden, müssen Sie bei jeder Änderung der Anzahl der Knoten in Ihrem Memcached-Cluster die Endpunkte in Ihrer Anwendung aktualisieren.

Vertikale Skalierung von Memcached

Wenn Sie einen Memcached-Cluster aufwärts oder abwärts skalieren, müssen Sie einen neuen Cluster erstellen. Memcached-Cluster sind zunächst leer, bis sie von der Anwendung befüllt werden.

Important

Vergewissern Sie sich beim Skalieren zu einem kleineren Knotentyp davon, dass der Speicher des kleineren Knotentyps für Ihre Daten und den Overhead ausreicht. Weitere Informationen finden Sie unter [Auswahl der Cache-Knotengröße](#).

Themen

- [Vertikale Skalierung von Memcached \(Konsole\)](#)
- [Vertikale Skalierung von Memcached \(AWS CLI\)](#)
- [Vertikale Skalierung von Memcached \(ElastiCache API\)](#)

Vertikale Skalierung von Memcached (Konsole)

Das folgende Verfahren führt Sie durch die vertikale Skalierung Ihres Clusters mithilfe der ElastiCache Konsole.

So skalieren Sie einen Memcached-Cluster vertikal (Konsole)

1. Erstellen Sie einen neuen Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Finden der Endpunkte eines Clusters \(Konsole\)](#).
3. Löschen Sie den alten Cluster. Weitere Informationen finden Sie unter [Löschen eines neuen Knotens in Memcached](#).

Vertikale Skalierung von Memcached (AWS CLI)

Nachfolgend wird beschrieben, wie Sie Ihren Memcached-Cache-Cluster mit der AWS CLI vertikal skalieren.

So skalieren Sie einen Memcached-Cache-Cluster vertikal (AWS CLI)

1. Erstellen Sie einen neuen Cache-Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Cluster erstellen mit dem CLI](#).
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cluster-Endpunkte. Weitere Informationen finden Sie unter [Suchen von Endpunkten \(AWS CLI\)](#).
3. Löschen Sie den alten Cache-Cluster. Weitere Informationen finden Sie unter [Verwendung der AWS CLI](#).

Vertikale Skalierung von Memcached (ElastiCache API)

Das folgende Verfahren führt Sie durch die vertikale Skalierung Ihres Memcached-Cache-Clusters mithilfe von ElastiCache API

So skalieren Sie einen Memcache-Cache-Cluster vertikal () ElastiCache API

1. Erstellen Sie einen neuen Cache-Cluster mit dem neuen Knotentyp. Weitere Informationen finden Sie unter [Einen Cluster \(ElastiCache API\) erstellen](#)
2. Aktualisieren Sie in Ihrer Anwendung die Endpunkte auf die neuen Cache-Cluster-Endpunkte. Weitere Informationen finden Sie unter [Endpunkte finden \(\) ElastiCache API](#).
3. Löschen Sie den alten Cache-Cluster. Weitere Informationen finden Sie unter [Verwenden der ElastiCache-API](#).

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 AWSBewährte 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`
- `CreateCacheParameterGroup`
- `CreateCacheSecurityGroup`
- `CreateCacheSubnetGroup`

- `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

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](#).

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 memcached
```



```
--tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- Hinzufügen von Tags zu einem Serverless-Cache

```
aws elasticache add-tags-to-resource \
--resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- 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 memcached \
--tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

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. Verweigern der Aktion `CreateCacheCluster`, wenn das Anforderungs-Tag `Project` fehlt oder nicht `Dev`, `QA` oder `Prod` entspricht.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*",
        "arn:aws:elasticache:*:*:securitygroup:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    },
    {
      "Effect": "Allow",

```

```
"Action": [
    "elasticache:CreateCacheCluster",
    "elasticache:AddTagsToResource"
],
"Resource": "arn:aws:elasticache:*:*:cluster:*",
"Condition": {
    "StringEquals": {
        "aws:RequestTag/Project": [
            "Dev",
            "Prod",
            "QA"
        ]
    }
}
}
```

Weitere Informationen zu Bedingungsschlüsseln finden Sie unter [Verwenden von Bedingungsschlüssel](#).

Überwachung von Kosten mit Kostenzuordnungs-Tags

Wenn Sie Ihren Ressourcen in Amazon Kostenzuordnungs-Tags hinzufügen ElastiCache, können Sie die Kosten verfolgen, indem Sie die Ausgaben auf Ihren Rechnungen nach Ressourcen-Tag-Werten gruppieren.

Ein ElastiCache Kostenzuweisungs-Tag ist ein Schlüssel-Wert-Paar, das Sie definieren und einer Ressource zuordnen. ElastiCache Bei 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 Ihrer eigenen Kostenstruktur entspricht. Melden Sie sich dazu an, um Ihre AWS Kontorechnung

mit den Tag-Schlüsselwerten zu erhalten. Um dann die Kosten kombinierter Ressourcen anzuzeigen, organisieren Sie Ihre Fakturierungsinformationen nach Ressourcen mit gleichen Tag-Schlüsselwerten. Beispielsweise können Sie mehrere Ressourcen mit einem bestimmten Anwendungsnamen markieren und dann Ihre Fakturierungsinformationen so organisieren, dass Sie die Gesamtkosten dieser Anwendung über mehrere Services hinweg sehen können.

Sie können auch Tags miteinander kombinieren, um Kosten detaillierter zu verfolgen. Um beispielsweise Ihre Kosten für Services nach Region zu verfolgen, könnten Sie die Tag-Schlüssel `Service` und `Region` verwenden. Für eine Ressource lauten die Werte möglicherweise `ElastiCache` und `Asia Pacific (Singapore)` und für eine andere Ressource lauten sie `ElastiCache` und `Europe (Frankfurt)`. Sie können dann Ihre ElastiCache Gesamtkosten nach Regionen aufgeschlüsselt sehen. Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#) im AWS Billing -Benutzerhandbuch.

Sie können ElastiCache Memcached-Clustern Tags für die Kostenzuweisung hinzufügen. Wenn Sie ein Tag hinzufügen, auflisten, ändern, kopieren oder entfernen, wird die Operation nur auf die angegebenen Cluster angewendet.

Eigenschaften von ElastiCache Kostenzuweisungs-Tags

- Kostenzuweisungs-Tags werden auf ElastiCache Ressourcen angewendet, die in CLI- und API-Vorgängen als ARN angegeben sind. Der Ressourcentyp ist ein „Cluster“.

Beispiel-ARN: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Memcached: Tags werden nur auf Cluster angewendet.

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- Der Tag-Schlüssel ist der erforderliche Name des Tags. Der Zeichenfolgenwert kann aus 1 bis 128 Unicode-Zeichen bestehen. Ihm darf kein `aws:` als Präfix vorangestellt werden. Die Zeichenfolge darf nur Unicode-Zeichen, Ziffern, Leerzeichen, Unterstriche (`_`), Punkte (`.`), Doppelpunkte (`:`), Backslashes (`\`), Gleichheitszeichen (`=`), Pluszeichen (`+`), Trennstriche (`-`) oder At-Zeichen (`@`) enthalten.
- Der Tag-Wert ist der optionale Wert des Tags. Der Zeichenfolgenwert kann aus 1 bis 256 Unicode-Zeichen bestehen. Ihm darf kein `aws:` als Präfix vorangestellt werden. Die Zeichenfolge darf nur

Unicode-Zeichen, Ziffern, Leerzeichen, Unterstriche (_), Punkte (.), Doppelpunkte (:), Backslashes (\), Gleichheitszeichen (=), Pluszeichen (+), Trennstriche (-) oder At-Zeichen (@) enthalten.

- Eine ElastiCache Ressource kann maximal 50 Tags haben.
- Die Werte innerhalb eines Tag-Satzes müssen nicht eindeutig sein. Beispiel: In einem Tag-Satz könnten die Schlüssel `Service` und `Application` beide den Wert `ElastiCache` besitzen.

AWS wendet Ihren Tags keine semantische Bedeutung an. Tags werden ausschließlich als Zeichenketten interpretiert. AWS setzt nicht automatisch irgendwelche Tags für eine ElastiCache Ressource.

Verwaltung Ihrer Kostenzuweisungs-Tags mithilfe der AWS CLI

Sie können die verwenden, AWS CLI um Kostenzuordnungs-Tags hinzuzufügen, zu ändern oder zu entfernen.

Kostenzuweisungs-Tags werden auf ElastiCache (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`

Beispiel-ARN: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

Themen

- [Auflisten von Tags mithilfe von AWS CLI](#)
- [Hinzufügen von Tags mit dem AWS CLI](#)
- [Ändern von Tags mit dem AWS CLI](#)
- [Entfernen von Tags mit dem AWS CLI](#)

Auflisten von Tags mithilfe von AWS CLI

Sie können die verwenden AWS CLI , um Tags für eine vorhandene ElastiCache Ressource aufzulisten, indem Sie die [list-tags-for-resource](#) Operation verwenden.

Der folgende Code verwendet die AWS CLI , um die Tags auf dem Memcached-Cluster `my-cluster` in der Region `us-west-2` aufzulisten.

Für Linux, macOS oder Unix:

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Für Windows:

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Die Ausgabe dieser Operation sieht in etwa folgendermaßen aus und besteht aus einer Liste aller Tags für die Ressource.

```
{  
  "TagList": [  
    {  
      "Value": "10110",  
      "Key": "CostCenter"  
    },  
    {  
      "Value": "EC2",  
      "Key": "Service"  
    }  
  ]  
}
```

Wenn die Ressource keine Tags enthält, ist die Ausgabe leer. `TagList`

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

Weitere Informationen finden Sie unter AWS CLI für ElastiCache [list-tags-for-resource](#).

Hinzufügen von Tags mit dem AWS CLI

Sie können das verwenden AWS CLI , um einer vorhandenen ElastiCache Ressource mithilfe der [add-tags-to-resource](#) CLI-Operation Tags 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 AWS CLI , um die Schlüssel Service und Region mit den Werten `elasticache` `us-west-2` jeweils zum Region `us-west-2` hinzuzufügen.

Für Linux, macOS oder Unix:

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \  
  --tags Key=Service,Value=elasticache \  
         Key=Region,Value=us-west-2
```

Für Windows:

```
aws elasticache add-tags-to-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^  
  --tags Key=Service,Value=elasticache ^  
         Key=Region,Value=us-west-2
```

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 für. AWS CLI ElastiCache [add-tags-to-resource](#)

Sie können den auch verwenden AWS CLI , um einem Cluster Tags hinzuzufügen, wenn Sie mithilfe des Vorgangs einen neuen Cluster erstellen [create-cache-cluster](#). Sie können keine Tags hinzufügen,

wenn Sie einen Cluster mithilfe der ElastiCache Managementkonsole erstellen. Nachdem der Cluster erstellt wurde, können Sie über die Konsole Tags zum Cluster hinzufügen.

Ändern von Tags mit dem AWS CLI

Sie können den verwenden AWS CLI , um die Tags auf einem ElastiCache (Memcached-) Cluster zu 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 mit dem AWS CLI

Sie können den verwenden AWS CLI , um Tags aus einem vorhandenen ElastiCache (Memcached-) Cluster zu entfernen, indem Sie den [remove-tags-from-resource](#) Vorgang verwenden.

Der folgende Code verwendet die AWS CLI , um die Tags mit den Schlüsseln `Service` und `Region` vom der Region `us-west-2` zu entfernen.

Für Linux, macOS oder Unix:

```
aws elasticache remove-tags-from-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \  
  --tag-keys PM Service
```

Für Windows:

```
aws elasticache remove-tags-from-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^  
  --tag-keys PM Service
```

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 für. AWS CLI ElastiCache [remove-tags-from-resource](#)

Verwaltung Ihrer Kostenzuweisungs-Tags mithilfe der ElastiCache API

Sie können die ElastiCache API verwenden, um Kostenzuweisungs-Tags hinzuzufügen, zu ändern oder zu entfernen.

Kostenzuweisungs-Tags werden ElastiCache für 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

- [Tags mithilfe der API auflisten ElastiCache](#)
- [Hinzufügen von Tags mithilfe der API ElastiCache](#)
- [Ändern von Tags mithilfe der ElastiCache API](#)
- [Tags mithilfe der ElastiCache API entfernen](#)

Tags mithilfe der API auflisten ElastiCache

Sie können die ElastiCache API verwenden, um Tags für eine vorhandene Ressource aufzulisten, indem Sie den [ListTagsForResource](#) Vorgang verwenden.

Der folgende Code verwendet die ElastiCache API, um die Tags auf der Ressource `my-cluster` in der Region `us-west-2` aufzulisten.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ListTagsForResource
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Version=2015-02-02
&Timestamp=20150202T192317Z
&X-Amz-Credential=<credential>
```

Hinzufügen von Tags mithilfe der API ElastiCache

Sie können die ElastiCache API verwenden, um einem vorhandenen ElastiCache Cluster Tags hinzuzufügen, indem Sie den [AddTagsToResource](#) 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 verwendet die ElastiCache API, um die Schlüssel `Service` und `Region` mit den Werten `elasticache us-west-2` jeweils zur Ressource `my-cluster` 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  
&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 [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.

Tags mithilfe der ElastiCache API entfernen

Sie können die ElastiCache API verwenden, um Tags aus einem vorhandenen ElastiCache (Memcached-) Cluster zu entfernen, indem Sie den [RemoveTagsFromResource](#) Vorgang verwenden.

Der folgende Code verwendet die ElastiCache API, um die Tags mit den Schlüsseln `Service` und `Region` vom Region `us-west-2` zu entfernen.

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=RemoveTagsFromResource  
&ResourceName=arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster  
&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](#)
- [Sicherheitssäule von Amazon ElastiCache Well-Architected Lens](#)
- [Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens](#)
- [Säule der Leistung und Effizienz von Amazon ElastiCache Well-Architected Lens](#)
- [Säule zur ElastiCache Kostenoptimierung bei Amazon Well-Architected Lens](#)

Säule Operational Excellence von Amazon ElastiCache Well-Architected Lens

Die Säule „Operational Excellence“ konzentriert sich auf den Betrieb und die Überwachung von Systemen, um einen Mehrwert für das Unternehmen zu schaffen, sowie auf die kontinuierliche Verbesserung von Prozessen und Verfahren. Zu den wichtigsten Themen gehören die Automatisierung von Änderungen, die Reaktion auf Ereignisse und die Definition von Standards für die Verwaltung des täglichen Betriebs.

Themen

- [OE 1: Wie verstehen Sie Warnmeldungen und Ereignisse, die von Ihrem ElastiCache Cluster ausgelöst werden, und wie reagieren Sie darauf?](#)
- [OE 2: Wann und wie skalieren Sie Ihre vorhandenen Cluster? ElastiCache](#)
- [OE 3: Wie verwalten Sie Ihre ElastiCache Cluster-Ressourcen und pflegen Ihren Cluster up-to-date?](#)
- [OE 4: Wie verwalten Sie die Verbindungen der Kunden zu Ihren ElastiCache Clustern?](#)
- [OE 5: Wie werden ElastiCache Komponenten für einen Workload bereitgestellt?](#)
- [OE 6: Wie planen und minimieren Sie Ausfälle?](#)
- [OE 7: Wie behebt man OSS Redis-Engine-Ereignisse?](#)

OE 1: Wie verstehen Sie Warnmeldungen und Ereignisse, die von Ihrem ElastiCache Cluster ausgelöst werden, und wie reagieren Sie darauf?

Einführung auf Fragenebene: Wenn Sie ElastiCache Cluster betreiben, können Sie optional Benachrichtigungen und Warnmeldungen erhalten, wenn bestimmte Ereignisse eintreten.

ElastiCache protokolliert standardmäßig [Ereignisse](#), die sich auf Ihre Ressourcen beziehen, z. B. Failover, Knotenaustausch, Skalierungsvorgänge, geplante Wartungsarbeiten und mehr. Jedes Ereignis enthält das Datum und die Uhrzeit, den Quellnamen und den Quelltyp sowie eine Beschreibung.

Vorteil auf Fragenebene: Wenn Sie in der Lage sind, die zugrundeliegenden Ursachen für die Ereignisse, die von Ihrem Cluster generierte Warnmeldungen auslösen, zu verstehen und zu verwalten, können Sie effektiver arbeiten und angemessen auf Ereignisse reagieren.

- [Erforderlich] Überprüfen Sie die Ereignisse, die von ElastiCache auf der ElastiCache Konsole (nach Auswahl Ihrer Region) oder mithilfe des Befehls [describe-events](#) von [Amazon Command Line Interface](#) (AWS CLI) und dem generiert wurden. [ElastiCache API](#) Konfigurieren Sie ElastiCache das Senden von Benachrichtigungen für wichtige Cluster-Ereignisse mithilfe von Amazon Simple Notification Service (AmazonSNS). Wenn Sie Amazon SNS mit Ihren Clustern verwenden, können Sie bei ElastiCache Ereignissen programmgesteuert Maßnahmen ergreifen.
- Es gibt zwei große Ereigniskategorien: aktuelle und geplante Ereignisse. Die Liste der aktuellen Ereignisse umfasst: Erstellung und Löschung von Ressourcen, Skalierungsvorgänge, Failover, Neustart des Knotens, erstellter Snapshot, Änderung der Cluster-Parameter, Erneuerung des CA-Zertifikats, Fehlerereignisse (Cluster-Bereitstellungsfehler - VPC oder -, Skalierungsfehler ENI - - und Snapshot-Fehler). ENI Die Liste der geplanten Ereignisse umfasst: Knoten, dessen Austausch während des Wartungsfensters geplant ist, und Knotenaustausch, der verschoben wurde.
- Auch wenn Sie auf einige dieser Ereignisse möglicherweise nicht sofort reagieren müssen, ist es wichtig, sich zunächst alle Fehlerereignisse anzusehen:
 - ElastiCache:AddCacheNodeFailed
 - ElastiCache:CacheClusterProvisioningFailed
 - ElastiCache:CacheClusterScalingFailed
 - ElastiCache:CacheNodesRebooted
 - ElastiCache: SnapshotFailed (Nur RedisOSS)
- [Ressourcen]:
 - [ElastiCache SNS Amazon-Benachrichtigungen verwalten](#)

- [Ereignisbenachrichtigungen und Amazon SNS](#)

- [Am besten] Nutzen Sie AWS Produkt- und Servicefunktionen wie SNS Lambda Functions, um Reaktionen auf Ereignisse zu automatisieren. Halten Sie sich an bewährte Methoden, indem Sie kleine, häufige, umkehrbare Änderungen als Code vornehmen, um Ihre Operationen im Laufe der Zeit weiterzuentwickeln. Sie sollten CloudWatch Amazon-Metriken verwenden, um Ihre Cluster zu überwachen.

[Ressourcen]: [Überwachen Sie ElastiCache \(RedisOSS\) \(Cluster-Modus deaktiviert\), lesen Sie Replica-Endpunkte mit AWS Lambda, Amazon Route 53 und Amazon SNS für einen Anwendungsfall](#), der Lambda und verwendet. SNS

OE 2: Wann und wie skalieren Sie Ihre vorhandenen Cluster? ElastiCache

Einführung auf Fragenebene: Die richtige Größe Ihres ElastiCache Clusters ist ein Balanceakt, der bei jeder Änderung der zugrunde liegenden Workload-Typen bewertet werden muss. Ihr Ziel ist es, mit der richtigen Größe für Ihren Workload zu arbeiten.

Vorteil auf Fragenebene: Eine Überlastung Ihrer Ressourcen kann zu einer erhöhten Latenz und einer insgesamt verringerten Leistung führen. Eine Unterauslastung kann andererseits zu einer Überbereitstellung von Ressourcen bei nicht optimaler Kostenoptimierung führen. Durch die richtige Dimensionierung Ihrer Umgebungen können Sie ein Gleichgewicht zwischen Leistungseffizienz und Kostenoptimierung erreichen. Um eine Über- oder Unterauslastung Ihrer Ressourcen zu beheben, ElastiCache können Sie in zwei Dimensionen skalieren. Sie können vertikal skalieren, indem Sie die Knotenkapazität erhöhen oder verringern. Sie können auch horizontal skalieren, indem Sie Knoten hinzufügen und entfernen.

- [Erforderlich] CPU und Netzwerküberlastung auf primären Knoten sollten behoben werden, indem die Lesevorgänge ausgelagert und auf Replikatknoten umgeleitet werden. Verwenden Sie Replikatknoten für Lesevorgänge, um die Auslastung des Primärknotens zu reduzieren. Dies kann in Ihrer OSS Redis-Client-Bibliothek konfiguriert werden, indem Sie eine Verbindung zum ElastiCache Reader-Endpunkt herstellen, wenn der Clustermodus deaktiviert ist, oder indem Sie den OSS READONLY Redis-Befehl für den Clustermodus aktivieren.

[Ressourcen]:

- [Ermitteln von Verbindungsendpunkten](#)
- [Richtige Clustergröße](#)
- [Redis-Befehl OSS READONLY](#)

- [Erforderlich] Überwachen Sie die Nutzung kritischer Clusterressourcen wie CPU Arbeitsspeicher und Netzwerk. Die Auslastung dieser spezifischen Clusterressourcen muss nachverfolgt werden, um eine Entscheidung für eine Skalierung und die Art des Skalierungsvorgangs treffen zu können. Wenn der Clustermodus ElastiCache (RedisOSS) deaktiviert ist, können Primär- und Replikatknoten vertikal skaliert werden. Replikatknoten können auch horizontal von 0 auf 5 Knoten skaliert werden. Wenn der Clustermodus aktiviert ist, gilt dasselbe für jeden Shard Ihres Clusters. Darüber hinaus können Sie die Anzahl der Shards erhöhen oder reduzieren.

[Ressourcen]:

- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) unter Verwendung von Amazon CloudWatch](#)
- [Skalierung von ElastiCache \(Redis OSS -\) Clustern](#)
- [Skalierung ElastiCache für Memcached-Cluster](#)
- [Am besten] Wenn Sie Trends im Zeitverlauf überwachen, können Sie Workload-Änderungen erkennen, die bei punktueller Überwachung unbemerkt bleiben würden. Um längerfristige Trends zu erkennen, verwenden Sie CloudWatch Metriken, um nach längeren Zeiträumen zu suchen. Die Erkenntnisse aus der Beobachtung von CloudWatch Kennzahlen über längere Zeiträume sollten als Grundlage für Ihre Prognose zur Auslastung der Cluster-Ressourcen dienen. CloudWatch Datenpunkte und Metriken sind für bis zu 455 Tage verfügbar.

[Ressourcen]:

- [Überwachung ElastiCache \(RedisOSS\) mit Metriken CloudWatch](#)
- [Überwachung von Memcached mit Metriken CloudWatch](#)
- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) unter Verwendung von Amazon CloudWatch](#)
- [Optimal] Wenn Ihre ElastiCache Ressourcen damit erstellt werden, empfiehlt CloudFormation es sich, Änderungen mithilfe von CloudFormation Vorlagen vorzunehmen, um die betriebliche Konsistenz zu wahren und unverwaltete Konfigurationsänderungen und Stack-Drifts zu vermeiden.

[Ressourcen]:

- [ElastiCache Referenz zum Ressourcentyp für CloudFormation](#)
- [Am besten] Automatisieren Sie Ihre Skalierungsvorgänge mithilfe von Cluster-Betriebsdaten und definieren Sie Schwellenwerte für CloudWatch die Einrichtung von Alarmen. Verwenden Sie CloudWatch Events und Simple Notification Service (SNS), um Lambda-Funktionen auszulösen und auszuführen und Ihre Cluster automatisch ElastiCache API zu skalieren. Ein Beispiel wäre,

Ihrem Cluster einen Shard hinzuzufügen, wenn die Metrik `EngineCPUUtilization` über einen längeren Zeitraum 80 % erreicht. Eine weitere Option wäre die Verwendung von `DatabaseMemoryUsedPercentages` für einen speicherbasierten Schwellenwert.

[Ressourcen]:

- [Amazon CloudWatch Alarms verwenden](#)
- [Was sind CloudWatch Amazon-Events?](#)
- [Verwendung AWS Lambda mit Amazon Simple Notification Service](#)
- [ElastiCacheAPIReferenz](#)

OE 3: Wie verwalten Sie Ihre ElastiCache Cluster-Ressourcen und pflegen Ihren Cluster up-to-date?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, ist es wichtig, dass Sie in der Lage sind, all Ihre Ressourcen zu lokalisieren und zu identifizieren. ElastiCache Bei der Einführung neuer Anwendungsfunktionen müssen Sie für eine Symmetrie der Cluster-Versionen in all Ihren ElastiCache Umgebungstypen sorgen: Entwicklung, Testen und Produktion. Mithilfe von Ressourcenattributen können Sie Umgebungen für unterschiedliche betriebliche Ziele trennen, z. B. bei der Einführung neuer Funktionen und der Aktivierung neuer Sicherheitsmechanismen.

Vorteil auf Fragenebene: Die Trennung Ihrer Entwicklungs-, Test- und Produktionsumgebungen ist die beste betriebliche Methode. Es ist auch eine bewährte Methode, dass auf Ihren Clustern und Knoten in allen Umgebungen die neuesten Softwarepatches installiert werden, wobei wohlverstandene und dokumentierte Prozesse verwendet werden. Durch die Nutzung systemeigener ElastiCache Funktionen kann sich Ihr Entwicklungsteam auf die Erreichung der Geschäftsziele konzentrieren und nicht auf die ElastiCache Wartung.

- [Am besten] Verwenden Sie die neueste verfügbare Engine-Version und installieren Sie die Self-Service-Updates, sobald sie verfügbar sind. ElastiCache aktualisiert die zugrunde liegende Infrastruktur automatisch während des angegebenen Wartungsfensters des Clusters. Die in Ihren Clustern laufenden Knoten werden jedoch über Self-Service-Updates aktualisiert. Es gibt zwei Arten von Updates: Sicherheitspatches oder kleinere Software-Updates. Stellen Sie sicher, dass Sie den Unterschied zwischen den Patch-Typen verstehen und wissen, wann diese angewendet werden.

[Ressourcen]:

- [Self-Service-Updates bei Amazon ElastiCache](#)

- [Hilfeseite für Amazon ElastiCache Managed Maintenance and Service Updates](#)
- [Am besten] Organisieren Sie Ihre ElastiCache Ressourcen mithilfe von Tags. Verwenden Sie Tags für Replikationsgruppen und nicht für einzelne Knoten. Sie können Tags so konfigurieren, dass sie angezeigt werden, wenn Sie Ressourcen abfragen, und Sie können Tags verwenden, um Suchen durchzuführen und Filter anzuwenden. Sie sollten Ressourcengruppen verwenden, um auf einfache Weise Sammlungen von Ressourcen zu erstellen und zu verwalten, die gemeinsame Tag-Sätze verwenden.

[Ressourcen]:

- [Bewährte Methoden zum Taggen](#)
- [ElastiCache Referenz zum Ressourcentyp für CloudFormation](#)
- [Parametergruppen](#)

OE 4: Wie verwalten Sie die Verbindungen der Kunden zu Ihren ElastiCache Clustern?

Einführung auf Fragenebene: Wenn Sie in großem Maßstab arbeiten, müssen Sie verstehen, wie Ihre Kunden mit dem ElastiCache Cluster kommunizieren, um die betrieblichen Aspekte Ihrer Anwendung (z. B. Reaktionszeiten) verwalten zu können.

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 aufgrund der asynchronen Natur der Redis-Replikation nicht mehr in der Lage sind, einen Schlüssel unmittelbar nach dem Schreiben zu lesen, wenn Sie die Schreibvorgänge von den Lesevorgängen trennen. OSS Der WAIT Befehl kann genutzt werden, um die Datensicherheit in der Praxis zu verbessern und Replikate zu zwingen, Schreibvorgänge zu bestätigen, bevor sie auf Clients antworten, was wiederum zu Leistungseinbußen führt. Die Verwendung von Replikatknoten für Lesevorgänge kann in Ihrer ElastiCache (Redis-OSS) Clientbibliothek konfiguriert werden, indem der ElastiCache Reader-Endpunkt für den Clustermodus deaktiviert ist. Wenn der Clustermodus aktiviert ist, verwenden Sie den Befehl ElastiCache (RedisOSS). READONLY Für viele der ElastiCache (Redis-OSS) Clientbibliotheken READONLY ist ElastiCache (RedisOSS) standardmäßig oder über eine Konfigurationseinstellung implementiert.

[Ressourcen]:

- [Ermitteln von Verbindungsendpunkten](#)
- [READONLY](#)
- [Erforderlich] Verwenden Sie Verbindungspooling. Das Herstellen einer TCP Verbindung ist sowohl auf der Client- als auch auf der Serverseite mit CPU Zeitaufwand verbunden, und das Pooling ermöglicht es Ihnen, die Verbindung wiederzuverwenden. TCP

Um den Verbindungsaufwand zu reduzieren, sollten Sie Verbindungspooling verwenden. Mit einem Pool von Verbindungen kann Ihre Anwendung Verbindungen „nach Belieben“ wiederverwenden und freigeben, ohne dass Kosten für den Verbindungsaufbau anfallen. Sie können das Verbindungspooling über Ihre ElastiCache (Redis-OSS) Clientbibliothek (sofern unterstützt) implementieren, wobei ein Framework für Ihre Anwendungsumgebung verfügbar ist, oder es von Grund auf neu erstellen.

- [Am besten] Stellen Sie sicher, dass das Socket-Timeout des Clients auf mindestens eine Sekunde eingestellt ist (im Vergleich zur typischen Standardeinstellung „Keine“ bei verschiedenen Clients).
 - Wenn Sie den Timeout-Wert zu niedrig einstellen, kann dies zu möglichen Timeouts führen, wenn die Serverlast hoch ist. Eine zu hohe Einstellung kann dazu führen, dass Ihre Anwendung lange braucht, um Verbindungsprobleme zu erkennen.
 - Kontrollieren Sie das Volumen neuer Verbindungen, indem Sie Verbindungspooling in Ihrer Client-Anwendung implementieren. Dies reduziert die Latenz und die CPU Auslastung, die zum Öffnen und Schließen von Verbindungen und zum Ausführen eines TLS Handshakes erforderlich sind, falls dies auf dem TLS Cluster aktiviert ist.

[Ressourcen]: [Konfiguration ElastiCache \(RedisOSS\) für höhere Verfügbarkeit](#)

- [Gut] Der Einsatz von Pipelining (sofern Ihre Anwendungsfälle dies zulassen) kann die Leistung erheblich steigern.
 - Mit Pipelining reduzieren Sie die RTT Round-Trip-Zeit () zwischen Ihren Anwendungsclients und dem Cluster. Neue Anfragen können verarbeitet werden, auch wenn der Client die vorherigen Antworten noch nicht gelesen hat.
 - Mit Pipelining können Sie mehrere Befehle an den Server senden, ohne auf Antworten/ Bestätigungen warten zu müssen. Der Nachteil von Pipelining ist, dass, wenn Sie irgendwann alle Antworten in Massen abrufen, möglicherweise ein Fehler aufgetreten ist, den Sie erst am Ende erkennen können.
 - Implementieren Sie Methoden, um Anfragen erneut zu versuchen, wenn ein Fehler zurückgegeben wird, der die fehlerhafte Anfrage auslöst.

[Ressourcen]: [Pipelining](#)

OE 5: Wie werden ElastiCache Komponenten für einen Workload bereitgestellt?

Einführung auf Fragenebene: ElastiCache Umgebungen können manuell über die AWS Konsole oder programmgesteuert über APIs, Toolkits usw. bereitgestellt werden. CLI Bewährte Methoden für Operational Excellence sehen vor, Bereitstellungen möglichst mithilfe von Code zu automatisieren. Darüber hinaus können ElastiCache Cluster entweder nach Arbeitslast isoliert oder zur Kostenoptimierung kombiniert werden.

Vorteil auf Frageebene: Durch die Auswahl des für Ihre ElastiCache Umgebungen am besten geeigneten Bereitstellungsmechanismus kann Operation Excellence im Laufe der Zeit verbessert werden. Es wird empfohlen, Operationen möglichst als Code auszuführen, um menschliche Fehler zu minimieren und die Wiederholbarkeit, Flexibilität und Reaktionszeit auf Ereignisse zu erhöhen.

Wenn Sie die Anforderungen an die Workload-Isolierung verstehen, können Sie entscheiden, ob Sie pro Workload dedizierte ElastiCache Umgebungen einrichten oder mehrere Workloads zu einzelnen Clustern oder Kombinationen daraus zusammenfassen möchten. Die Nachteile zu verstehen, kann dazu beitragen, ein Gleichgewicht zwischen Operational Excellence und Kostenoptimierung zu finden

- [Erforderlich] Machen Sie sich mit den verfügbaren Bereitstellungsoptionen vertraut und automatisieren Sie diese Verfahren ElastiCache, wann immer dies möglich ist. Zu den möglichen Automatisierungsmöglichkeiten gehören CloudFormationSDK, AWS CLI/und APIs.

[Ressourcen]:

- [Referenz zum ElastiCache Amazon-Ressourcentyp](#)
- [elasticache](#)
- [ElastiCache API Amazon-Referenz](#)
- [Erforderlich] Ermitteln Sie für alle Workloads den erforderlichen Grad der Clusterisolierung.
 - [Am besten]: Hohe Isolierung – eine 1:1-Zuordnung zwischen Workload und Cluster. Ermöglicht die feinste Kontrolle über den Zugriff, die Größe, Skalierung und Verwaltung von ElastiCache Ressourcen auf Workload-Basis.
 - [Besser]: Mittlere Isolierung – M:1 ist zweckmäßig isoliert, aber möglicherweise von mehreren Workloads gemeinsam genutzt (z. B. ein Cluster, der für das Zwischenspeichern von Workloads vorgesehen ist, und ein anderer, der für Messaging dediziert ist).

- [Gut]: Niedrige Isolierung – M:1 für alle Zwecke, vollständig gemeinsam genutzt. Empfohlen für Workloads, bei denen gemeinsamer Zugriff akzeptabel ist.

OE 6: Wie planen und minimieren Sie Ausfälle?

Einführung auf Fragenebene: Operational Excellence umfasst die Antizipation von Ausfällen durch regelmäßige „Pre-Mortem“-Übungen zur Identifizierung potenzieller Fehlerquellen, sodass sie behoben oder gemildert werden können. ElastiCache bietet ein FailoverAPI, das zu Testzwecken simulierte Knotenausfälle ermöglicht.

Vorteil auf Fragenebene: Wenn Sie Ausfallszenarien vorab testen, können Sie herausfinden, wie sie sich auf Ihren Workload auswirken. Dies ermöglicht ein sicheres Testen der Reaktionsverfahren und ihrer Wirksamkeit und macht Ihr Team mit deren Ausführung vertraut.

[Erforderlich] Führen Sie regelmäßig Failover-Tests in Entwickler-/Testkonten durch. [TestFailover](#)

OE 7: Wie behebt man OSS Redis-Engine-Ereignisse?

Einführung auf Fragenebene: Operational Excellence erfordert die Fähigkeit, sowohl Informationen auf Service- als auch auf Engine-Ebene zu untersuchen, um den Zustand und den Status Ihrer Cluster zu analysieren. ElastiCache (RedisOSS) kann OSS Redis-Engine-Protokolle sowohl an Amazon als auch an Amazon Kinesis CloudWatch Data Firehose senden.

Vorteil auf Fragenebene: Durch die Aktivierung von OSS Redis-Engine-Protokollen auf ElastiCache (Redis-OSS) Clustern erhalten Sie Einblicke in Ereignisse, die sich auf den Zustand und die Leistung von Clustern auswirken. OSSRedis-Engine-Protokolle liefern Daten direkt von der OSS Redis-Engine, die nicht über den Ereignismechanismus verfügbar sind. ElastiCache Durch sorgfältige Beobachtung sowohl der ElastiCache Ereignisse (siehe oben in OE-1) als auch der OSS Redis-Engine-Protokolle ist es möglich, bei der Fehlerbehebung eine Reihenfolge der Ereignisse sowohl aus der Sicht des ElastiCache Service als auch aus der Sicht der Redis-Engine zu bestimmen. OSS

- [Erforderlich] Stellen Sie sicher, dass die Protokollierungsfunktion der OSS Redis-Engine aktiviert ist. Diese Funktion ist ab ElastiCache (OSSRedis) 6.2 und neuer verfügbar. Dieser Vorgang kann während der Clustererstellung oder durch Änderung des Clusters nach der Erstellung erfolgen.
 - Stellen Sie fest, ob Amazon CloudWatch Logs oder Amazon Kinesis Data Firehose das geeignete Ziel für OSS Redis-Engine-Protokolle ist.
 - Wählen Sie ein geeignetes Zielprotokoll in CloudWatch oder Kinesis Data Firehose aus, um die Protokolle beizubehalten. Wenn Sie mehrere Cluster haben, sollten Sie für jeden Cluster ein

anderes Zielprotokoll verwenden, da Daten bei der Fehlerbehebung so besser isoliert werden können.

[Ressourcen]:

- Protokollzustellung: [Protokollzustellung](#)
- Ziele für die Protokollierung: [Amazon CloudWatch Logs](#)
- Einführung in Amazon CloudWatch Logs: [Was ist Amazon CloudWatch Logs?](#)
- Einführung in Amazon Kinesis Data Firehose: [Was ist Amazon Kinesis Data Firehose?](#)
- [Am besten] Wenn Sie Amazon CloudWatch Logs verwenden, sollten Sie Amazon CloudWatch Logs Insights nutzen, um das OSS Redis-Engine-Protokoll nach wichtigen Informationen abzufragen.

Erstellen Sie als Beispiel eine Abfrage für die CloudWatch Protokollgruppe, die die OSS Redis-Engine-Protokolle enthält, die Ereignisse mit dem Zeichen 'WARNING' zurückgeben, wie zum Beispiel: LogLevel

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[Ressourcen]: [Analysieren von Protokolldaten mit CloudWatch Logs Insights](#)

Sicherheitssäule von Amazon ElastiCache Well-Architected Lens

Die Säule der Sicherheit konzentriert sich auf den Schutz von Informationen und Systemen. Zu den wichtigsten Themen gehören die Vertraulichkeit und Integrität von Daten, die Ermittlung und Verwaltung von Berechtigungen mithilfe der Berechtigungsverwaltung, der Schutz von Systemen und die Einrichtung von Kontrollen zur Erkennung von Sicherheitsereignissen.

Themen

- [SEC 1: Welche Schritte ergreifen Sie, um den autorisierten Zugriff auf Daten zu ElastiCache kontrollieren?](#)
- [SEC 2: Benötigen Ihre Anwendungen zusätzliche Autorisierungen, die ElastiCache über netzwerkbasierter Kontrollen hinausgehen?](#)
- [SEC 3: Besteht das Risiko, dass Befehle versehentlich ausgeführt werden können und Datenverlust oder -ausfall verursachen?](#)

- [SEC 4: Wie gewährleisten Sie die Datenverschlüsselung im Ruhezustand mit ElastiCache](#)
- [SEC 5: Wie verschlüsselt man 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 ergreifen Sie, um den autorisierten Zugriff auf Daten zu ElastiCache kontrollieren?

Einführung auf Fragenebene: Alle ElastiCache Cluster sind für den Zugriff über Amazon Elastic Compute Cloud-Instances in einer VPC, serverlose Funktionen (AWS Lambda) oder Container (Amazon Elastic Container Service) konzipiert. Das am häufigsten anzutreffende Szenario ist der Zugriff auf einen ElastiCache Cluster von einer Amazon Elastic Compute Cloud-Instanz innerhalb derselben Amazon Virtual Private Cloud (Amazon Virtual Private Cloud). Bevor Sie von einer 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 Netzwerkzugang 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, wenn ein Cluster gestartet wird, muss eine Sicherheitsgruppe zugeordnet werden. Dadurch wird sichergestellt, dass Regeln für ein- und ausgehenden Datenverkehr für alle Knoten gelten, aus denen der Cluster besteht. Darüber hinaus ElastiCache ist es so konfiguriert, dass es ausschließlich in privaten Subnetzen bereitgestellt wird, sodass auf sie nur über das 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 auch kein Eingangspfad zu ElastiCache. Um dies zu aktivieren, konfigurieren Sie eine Regel für eingehende Nachrichten für die Sicherheitsgruppe, die Quell-IP-Adresse/-bereich, den TCP-Verkehr und den Port für Ihren ElastiCache Cluster (z. B. Standardport 6379 für ElastiCache (Redis OSS)). Es ist zwar möglich, eine sehr breite Palette von Eingangsquellen zuzulassen, z. B. alle Ressourcen innerhalb einer VPC (0.0.0.0/0), es wird jedoch empfohlen, die Regeln für eingehenden Datenverkehr so detailliert wie möglich zu definieren, z. B.

nur den eingehenden Zugriff auf Redis OSS-Clients zu autorisieren, die auf Amazon Amazon EC2 EC2-Instances laufen, die 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]AWS Identity and Access Management Richtlinien können Funktionen zugewiesen werden, die ihnen den Zugriff auf Daten ermöglichen. AWS Lambda ElastiCache Um diese Funktion zu aktivieren, erstellen Sie eine IAM-Ausführungsrolle mit der entsprechenden `AWSLambdaVPCLambdaAccessExecutionRole` Berechtigung und weisen Sie die Rolle dann der AWS Lambda Funktion zu.

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

SEC 2: Benötigen Ihre Anwendungen zusätzliche Autorisierungen, die ElastiCache über netzwerkbasierte Kontrollen hinausgehen?

Einführung auf Fragenebene: In Szenarien, in denen es notwendig ist, den Zugriff auf ElastiCache (Redis OSS) -Cluster auf individueller Client-Ebene einzuschränken oder zu kontrollieren, wird empfohlen, sich über den AUTH-Befehl ElastiCache (Redis OSS) zu authentifizieren. ElastiCache (Redis OSS) -Authentifizierungstoken mit optionaler Benutzer- und Benutzergruppenverwaltung ermöglichen es ElastiCache (Redis OSS), ein Passwort anzufordern, bevor Clients Befehle und Zugriffsschlüssel ausführen können, wodurch die Sicherheit der Datenebene verbessert wird.

Vorteil auf Frageebene: Um die Sicherheit Ihrer Daten zu gewährleisten, bietet ElastiCache (Redis OSS) Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten. Dazu gehört auch, dass Clients mit der Rollenbasierten Zugriffskontrolle (RBAC) AUTH oder AUTH-Token (Passwort) eine Verbindung herstellen müssen, bevor autorisierte Befehle ausgeführt werden. ElastiCache

- [Am besten] Definieren Sie für ElastiCache (Redis OSS) 6.x und höher Authentifizierungs- und Autorisierungskontrollen, indem Sie Benutzergruppen, Benutzer und Zugriffszeichenfolgen definieren. Weisen Sie Benutzer Benutzergruppen zu und weisen Sie Benutzergruppen dann Clustern zu. 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 OSS-Client verwenden, der TLS unterstützt, um RBAC nutzen zu können.

[Ressourcen]:

- [Anwenden von RBAC auf eine Replikationsgruppe für \(Redis OSS\) ElastiCache](#)
 - [Spezifizieren von Berechtigungen mithilfe einer Zugriffszeichenfolge](#)
 - [ACL](#)
 - [Unterstützte Versionen ElastiCache \(Redis OSS\)](#)
- [Optimal] Für ElastiCache (Redis OSS-) Versionen vor 6.x empfiehlt es sich, neben der Festlegung eines starken Tokens/Kennworts und der Einhaltung einer strengen Passworrichtlinie für ElastiCache (Redis OSS) AUTH auch die Rotation des Passworts/Tokens zu verwenden. ElastiCache kann bis zu zwei (2) Authentifizierungstoken gleichzeitig verwalten. Sie können den Cluster auch so ändern, dass explizit die Verwendung von Authentifizierungstoken erforderlich ist.

[Ressourcen]: [Ändern des AUTH-Tokens auf einem vorhandenen Cluster ElastiCache \(Redis OSS\)](#)

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 OSS-Befehlen, die sich nachteilig auf den Betrieb auswirken können, wenn sie versehentlich oder von böswilligen Akteuren ausgeführt werden. Diese Befehle können im Hinblick auf die Leistung und Datensicherheit unbeabsichtigte Folgen haben. Beispielsweise kann ein Entwickler 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 (Redis OSS) 5.0.3 haben Sie die Möglichkeit, bestimmte Befehle umzubenennen, die Ihre Arbeitslast stören könnten. Durch das Umbenennen der Befehle kann verhindert werden, dass diese versehentlich auf dem Cluster ausgeführt werden.

- [Erforderlich]

[Ressourcen]:

- [ElastiCache \(Redis OSS\) Version 5.0.3 \(veraltet, verwenden Sie Version 5.0.6\)](#)
- [Änderungen an den Parametern von Redis OSS 5.0.3](#)
- [Redis OSS-Sicherheit](#)

SEC 4: Wie gewährleisten Sie die Datenverschlüsselung im Ruhezustand mit ElastiCache

Einführung auf Fragenebene: Bei ElastiCache (Redis OSS) handelt es sich zwar um einen In-Memory-Datenspeicher, es ist jedoch möglich, alle Daten zu verschlüsseln, die im Rahmen der Standardoperationen des Clusters dauerhaft (im Speicher) 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 (Redis OSS) bietet optionale Verschlüsselung im Ruhezustand, um die Datensicherheit zu erhöhen.

- [Erforderlich] Die Verschlüsselung im Ruhezustand kann auf einem ElastiCache Cluster (Replikationsgruppe) nur aktiviert werden, wenn dieser erstellt wurde. Ein vorhandener Cluster kann nicht geändert werden, um mit der Verschlüsselung von Daten im Ruhezustand zu beginnen. Stellt standardmäßig die Schlüssel bereit und verwaltet ElastiCache sie, die bei der Verschlüsselung im Ruhezustand verwendet werden.

[Ressourcen]:

- [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. In Umgebungen mit strengeren Datensicherheitsumgebungen kann AWS Key Management Service (KMS) zur Selbstverwaltung der Customer Master Keys (CMK) verwendet werden. Durch die ElastiCache Integration mit sind Sie in der Lage AWS Key Management Service, die Schlüssel zu erstellen, zu besitzen und zu verwalten, die für die Verschlüsselung ruhender Daten für Ihren ElastiCache (Redis OSS) -Cluster verwendet werden.

[Ressourcen]:

- [Verwenden Sie vom Kunden verwaltete Schlüssel von AWS Key Management Service](#)
- [AWS Schlüsselverwaltungsdienst](#)
- [AWS -KMS-Konzepte](#)

SEC 5: Wie verschlüsselt man Daten bei der Übertragung mit? ElastiCache

Einführung auf Fragenebene: Es ist eine gängige Anforderung, zu verhindern, dass Daten während der Übertragung beschädigt werden. Dabei handelt es sich um Daten innerhalb von Komponenten eines verteilten Systems sowie zwischen Anwendungsclients und Clusterknoten. ElastiCache (Redis OSS) unterstützt diese Anforderung, indem es die Verschlüsselung von Daten ermöglicht, die zwischen Clients und Clustern sowie zwischen den Clusterknoten selbst übertragen werden. Instance-Typen der M6g- und R6g-Familien verfügen außerdem über eine ständig aktive In-Memory-Verschlüsselung.

Vorteil auf Frageebene: Die Verschlüsselung ElastiCache während der Übertragung durch Amazon ist eine optionale Funktion, mit der Sie die Sicherheit Ihrer Daten an den anfälligsten Stellen erhöhen können, wenn sie von einem Ort zum anderen übertragen werden.

- [Erforderlich] Die Verschlüsselung bei der Übertragung kann bei der Erstellung nur auf einem ElastiCache (Redis OSS) -Cluster (Replikationsgruppe) aktiviert werden. Bitte beachten Sie, dass die Implementierung der Verschlüsselung bei der Übertragung aufgrund der zusätzlichen Verarbeitung, die für die Ver-/Entschlüsselung von Daten erforderlich ist, Auswirkungen auf die Leistung hat. Um die Auswirkungen zu verstehen, wird empfohlen, Ihren Workload vor und nach der Aktivierung zu vergleichen. encryption-in-transit

[Ressourcen]:

- [Übersicht über die Verschlüsselung bei der Übertragung](#)

SEC 6: Wie schränken Sie den Zugriff auf Ressourcen der Steuerebene ein?

Einführung auf Fragenebene: IAM-Richtlinien und ARN ermöglichen detaillierte Zugriffskontrollen für ElastiCache (Redis OSS) und ermöglichen so eine strengere Kontrolle bei der Verwaltung der Erstellung, Änderung und Löschung von ElastiCache (Redis OSS) -Clustern.

Vorteil auf Frageebene: Die Verwaltung von ElastiCache Amazon-Ressourcen wie Replikationsgruppen, Knoten usw. kann auf AWS Konten beschränkt werden, die über spezifische Berechtigungen auf der Grundlage von IAM-Richtlinien verfügen, wodurch die Sicherheit und Zuverlässigkeit der Ressourcen verbessert wird.

- [Erforderlich] Verwalten Sie den Zugriff auf ElastiCache Amazon-Ressourcen, indem Sie AWS Benutzern bestimmte AWS Identity and Access Management Richtlinien zuweisen, sodass Sie genauer kontrollieren können, welche Konten welche Aktionen auf Clustern ausführen können.

[Ressourcen]:

- [Überblick über die Verwaltung der Zugriffsberechtigungen für Ihre Ressourcen ElastiCache](#)
- [Verwendung identitätsbasierter Richtlinien \(IAM-Richtlinien\) für Amazon ElastiCache](#)

SEC 7: Wie erkennen Sie Sicherheitsereignisse und wie reagieren Sie darauf?

Einführung auf Fragenebene: ElastiCache Exportiert bei der Implementierung mit aktiviertem RBAC 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 Produkte und Servicere Ressourcen dazu bei, Ihre gesamte Arbeitslast zu sichern, indem sie Bereitstellungen automatisieren und alle Aktionen und Änderungen für eine spätere Überprüfung/Prüfung protokollieren.

Vorteil auf Fragenebene: Durch die Überwachung von Ereignissen ermöglichen Sie Ihrem Unternehmen, gemäß Ihren Anforderungen, Richtlinien und Verfahren zu reagieren. Durch die Automatisierung der Überwachung und Reaktion auf diese Sicherheitsereignisse wird Ihre allgemeine Sicherheitslage gestärkt.

- [Erforderlich] Machen Sie sich mit den veröffentlichten CloudWatch Kennzahlen vertraut, die sich auf RBAC-Authentifizierungs- und Autorisierungsfehler beziehen.
 - AuthenticationFailures = Fehlgeschlagene Versuche, sich bei Redis OSS 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 OSS](#)
- [Am besten] Es wird empfohlen, Warnmeldungen und Benachrichtigungen für diese Metriken einzurichten und je nach Bedarf zu reagieren.

[Ressourcen]:

- [CloudWatch Amazon-Alarme verwenden](#)

- [Am besten] Verwenden Sie den Befehl `Redis OSS ACL LOG`, um weitere Informationen zu sammeln

[Ressourcen]:

- [ACL LOG](#)
- [Am besten] Machen Sie sich mit den Funktionen der AWS Produkte und Services im Zusammenhang mit der Überwachung, Protokollierung und Analyse von ElastiCache Bereitstellungen und Ereignissen vertraut

[Ressourcen]:

- [Protokollierung Amazon ElastiCache Amazon-API-Aufrufen mit AWS CloudTrail](#)
- [elasticache-redis-cluster-automatic-Backup-Check](#)
- [Überwachung der Nutzung mit Metrics CloudWatch](#)

Säule der Zuverlässigkeit von Amazon ElastiCache Well-Architected Lens

Themen

- [REL1: Wie unterstützen Sie Implementierungen von Hochverfügbarkeitsarchitekturen \(HA\)?](#)
- [REL2: Wie erreichen Sie Ihre Recovery Point Objectives \(RPOs\)? ElastiCache](#)
- [REL3: Wie unterstützen Sie die Anforderungen für die Notfallwiederherstellung \(DR\)?](#)
- [REL4: Wie können Sie Failovers effektiv planen?](#)
- [REL5: Sind Ihre ElastiCache Komponenten so konzipiert, dass sie skalierbar sind?](#)

REL1: Wie unterstützen Sie Implementierungen von Hochverfügbarkeitsarchitekturen (HA)?

Einführung auf Fragenebene: Wenn Sie die Hochverfügbarkeitsarchitektur von Amazon verstehen, können ElastiCache Sie bei Verfügbarkeitsereignissen in einem stabilen Zustand arbeiten.

Vorteil auf Fragenebene: Wenn Sie Ihre ElastiCache Cluster so gestalten, dass sie ausfallsicher sind, wird eine höhere Verfügbarkeit Ihrer Bereitstellungen gewährleistet. ElastiCache

- [Erforderlich] Ermitteln Sie das Maß an Zuverlässigkeit, das Sie für Ihren Cluster benötigen. ElastiCache Verschiedene Workloads haben unterschiedliche Resilienzstandards, von rein

kurzlebigen bis hin zu geschäftskritischen Workloads. Definieren Sie die Anforderungen für jede Art von Umgebung, die Sie betreiben, z. B. Entwicklung, Test und Produktion.

Caching-Engine: ElastiCache (Memcached) gegen ElastiCache (Redis) OSS

1. ElastiCache (Memcached) bietet keinen Replikationsmechanismus und wird hauptsächlich für kurzlebige Workloads verwendet.
 2. ElastiCache (RedisOSS) bietet HA-Funktionen, die im Folgenden beschrieben werden
- [Am besten] Verwenden Sie ElastiCache (RedisOSS) für Workloads, die HA erfordern, im Clustermodus mit mindestens zwei Replikaten pro Shard, auch für Workloads mit geringem Durchsatz, die nur einen Shard benötigen.
 1. Wenn der Clustermodus aktiviert ist, wird Multi-AZ automatisch aktiviert.

Multi-AZ minimiert Ausfallzeiten, indem automatische Failovers vom Primärknoten bis zu den Replikaten im Falle einer geplanten oder ungeplanten Wartung durchgeführt und AZ-Ausfälle minimiert werden.

2. Bei Shard-Workloads sorgen mindestens drei Shards für eine schnellere Wiederherstellung bei Failover-Ereignissen, da das OSS Redis-Cluster-Protokoll erfordert, dass die Mehrheit der Primärknoten verfügbar sein muss, um ein Quorum zu erreichen.
3. Richten Sie zwei oder mehr Replikate für die Verfügbarkeit ein.

Zwei Replikate bieten eine verbesserte Leseskalierbarkeit sowie Leseverfügbarkeit in Szenarien, in denen ein Replikat gewartet wird.

4. Verwenden Sie Graviton2-basierte Knotentypen (Standardknoten in den meisten Regionen).

ElastiCache (RedisOSS) hat die Leistung auf diesen Knoten optimiert. Dadurch erhalten Sie eine bessere Replikations- und Synchronisierungsleistung, was zu einer insgesamt verbesserten Verfügbarkeit führt.

5. Überwachen Sie und passen Sie die Größe an, um auf erwartete Verkehrsspitzen zu reagieren: Bei hoher Auslastung reagiert die ElastiCache (Redis-OSS) Engine möglicherweise nicht mehr, was sich auf die Verfügbarkeit auswirkt. `BytesUsedForCache` und `DatabaseMemoryUsagePercentage` sind gute Indikatoren für Ihre Speichernutzung, wohingegen `ReplicationLag` sie anhand Ihrer Schreibrate ein Indikator für den Zustand Ihrer Replikation sind. Sie können diese Metriken verwenden, um die Clusterskalierung auszulösen.
6. Stellen Sie die Stabilität auf der Clientseite sicher, indem Sie [API vor einem Produktions-Failover-Ereignis Tests mit dem Failover](#) durchführen.

[Ressourcen]:

- [Konfigurieren Sie ElastiCache \(Redis\) für eine höhere Verfügbarkeit OSS](#)
- [Hohe Verfügbarkeit mit Replikationsgruppen](#)

REL2: Wie erreichen Sie Ihre Recovery Point Objectives (RPOs)? ElastiCache

Einführung auf Fragenebene: Machen Sie sich mit der Arbeitslast vertraut RPO, um fundierte Entscheidungen über ElastiCache Sicherheits- und Wiederherstellungsstrategien treffen zu können.

Vorteil auf Fragenebene: Mit einer RPO Strategie vor Ort kann die Geschäftskontinuität im Fall von Notfallwiederherstellungsszenarien verbessert werden. Die Gestaltung Ihrer Sicherheits- und Wiederherstellungsrichtlinien kann Ihnen helfen, Ihre Recovery Point Objectives (RPO) für Ihre Daten zu erreichen. ElastiCache ElastiCache (RedisOSS) bietet Snapshot-Funktionen, die in Amazon S3 gespeichert werden, zusammen mit einer konfigurierbaren Aufbewahrungsrichtlinie. Diese Snapshots werden während eines definierten Backup-Fensters aufgenommen und vom Service automatisch verarbeitet. Wenn Ihr Workload zusätzliche Backup-Granularität erfordert, haben Sie die Möglichkeit, bis zu 20 manuelle Backups pro Tag zu erstellen. Manuell erstellte Backups unterliegen keiner Serviceaufbewahrungsrichtlinie und können auf unbestimmte Zeit aufbewahrt werden.

- [Erforderlich] Machen Sie sich mit Ihren ElastiCache Bereitstellungen vertraut und dokumentieren Sie sie. RPO
 - Beachten Sie, dass Memcached keine Backup-Prozesse anbietet.
 - Informieren Sie sich über die Funktionen der ElastiCache Backup- und Wiederherstellungsfunktionen.
- [Am besten] Etablieren Sie einen gut kommunizierten Prozess für die Sicherung Ihres Clusters.
 - Initiieren Sie bei Bedarf manuelle Backups.
 - Überprüfen Sie die Aufbewahrungsrichtlinien für automatische Backups.
 - Beachten Sie, dass manuelle Backups auf unbestimmte Zeit aufbewahrt werden.
 - Planen Sie Ihre automatischen Backups in Zeiten geringer Auslastung.
 - Führen Sie Backup-Operationen anhand von Lesereplikaten durch, um sicherzustellen, dass die Auswirkungen auf die Cluster-Leistung so gering wie möglich gehalten werden.
- [Gut] Nutzen Sie die Funktion für geplante Backups von ElastiCache , um Ihre Daten regelmäßig innerhalb eines bestimmten Zeitfensters zu sichern.
 - Testen Sie regelmäßig Wiederherstellungen aus Ihren Backups.

- [Ressourcen]:
 - [Redis OSS](#)
 - [Backup und Wiederherstellung für ElastiCache \(RedisOSS\)](#)
 - [Erstellen manueller Backups](#)
 - [Planen automatischer Backups](#)
 - [Cluster Backup und wiederherstellen ElastiCache \(RedisOSS\)](#)

REL3: Wie unterstützen Sie die Anforderungen für die Notfallwiederherstellung (DR)?

Einführung auf Fragenebene: Die Notfallwiederherstellung ist ein wichtiger Aspekt jeder Workload-Planung. ElastiCache (RedisOSS) bietet mehrere Optionen zur Implementierung von Disaster Recovery auf der Grundlage von Anforderungen an die Belastbarkeit von Workloads. Mit Amazon ElastiCache Global Datastore können Sie in Ihren ElastiCache (Redis-OSS) Cluster in einer Region schreiben und die Daten zum Lesen aus zwei anderen regionsübergreifenden Replikatclustern zur Verfügung stellen, wodurch Lesevorgänge mit niedriger Latenz und regionsübergreifende Disaster Recovery ermöglicht werden.

Vorteil auf Fragenebene: Durch das Nachvollziehen und die Planung verschiedener Notfallszenarien kann die Geschäftskontinuität gewährleistet werden. DR-Strategien müssen gegenüber Kosten, Leistungseinbußen und Datenverlustpotenzial abgewogen werden.

- [Erforderlich] Entwickeln und dokumentieren Sie DR-Strategien für all Ihre ElastiCache Komponenten auf der Grundlage der Workload-Anforderungen. ElastiCache ist insofern einzigartig, als einige Anwendungsfälle völlig kurzlebig sind und keine DR-Strategie erfordern, während andere am anderen Ende des Spektrums liegen und eine extrem robuste DR-Strategie erfordern. Alle Optionen müssen gegenüber Kostenoptimierung abgewogen werden – eine größere Resilienz erfordert eine umfassendere Infrastruktur.

Machen Sie sich mit den DR-Optionen vertraut, die auf regionaler und multiregionaler Ebene verfügbar sind.

- Multi-AZ-Bereitstellungen werden empfohlen, um sich vor AZ-Ausfällen zu schützen. Stellen Sie sicher, dass bei der Bereitstellung der Cluster-Modus in Multi-AZ-Architekturen aktiviert ist und mindestens 3 verfügbar sind. AZs
- Global Datastore wird als Schutz vor regionalen Ausfällen empfohlen.
- [Am besten] Aktivieren Sie Global Datastore für Workloads, die Resilienz auf regionaler Ebene erfordern.

- Planen Sie ein Failover auf die sekundäre Region für den Fall einer Verschlechterung der primären Region ein.
- Testen Sie den Failover-Prozess für mehrere Regionen, bevor Sie ein Failover in der Produktion durchführen.
- Überwachen Sie die `ReplicationLag`-Metrik, um die möglichen Auswirkungen von Datenverlusten bei Failover-Ereignissen zu verstehen.
- [Ressourcen]:
 - [Minimieren von Ausfällen](#)
 - [Regionsübergreifende Replikation mithilfe globaler Datenspeicher AWS](#)
 - [Wiederherstellen aus einem Backup mit optionaler Cluster-Größenanpassung](#)
 - [Minimierung der Ausfallzeiten in ElastiCache \(OSSRedis\) mit Multi-AZ](#)

REL4: Wie können Sie Failovers effektiv planen?

Einführung auf Fragenebene: Die Aktivierung von Multi-AZ mit automatischen Failovers ist eine bewährte Methode. ElastiCache In bestimmten Fällen ersetzt ElastiCache (RedisOSS) die Primärknoten im Rahmen von Servicebetrieben. Beispiele hierfür sind geplante Wartungsereignisse und der unwahrscheinliche Falle eines Ausfalls des Knotens oder eines Problems mit der Availability Zone. Erfolgreiche Failover hängen sowohl ElastiCache von beiden als auch von Ihrer Client-Bibliothekskonfiguration ab.

Vorteil auf Fragenebene: Wenn Sie die Best Practices für ElastiCache Failover in Verbindung mit Ihrer spezifischen ElastiCache (Redis-OSS) Clientbibliothek befolgen, können Sie potenzielle Ausfallzeiten bei Failover-Ereignissen minimieren.

- [Erforderlich] Verwenden Sie bei deaktiviertem Clustermodus Timeouts, damit Ihre Clients anhand der aktualisierten IP-Adresse des primären Endpunkts erkennen, ob die Verbindung zum alten Primärknoten getrennt und erneut eine Verbindung zum neuen Primärknoten hergestellt werden muss. Im aktivierten Clustermodus ist die Clientbibliothek dafür zuständig, Änderungen in der zugrundeliegenden Cluster-Topologie zu erkennen. Dies wird meistens durch Konfigurationseinstellungen in der ElastiCache (Redis-OSS) Clientbibliothek erreicht, mit denen Sie auch die Häufigkeit und die Methode der Aktualisierung konfigurieren können. Jede Clientbibliothek bietet eigene Einstellungen. Weitere Informationen finden Sie in der entsprechenden Dokumentation.

[Ressourcen]:

- [Minimierung der Ausfallzeiten in ElastiCache \(RedisOSS\) mit Multi-AZ](#)
- Informieren Sie sich über die Best Practices Ihrer ElastiCache (Redis OSS -) Client-Bibliothek.
- [Erforderlich] Erfolgreiche Failover hängen von einer fehlerfreien Replikationsumgebung zwischen dem Primärknoten und den Replikatknoten ab. Informieren Sie sich über die asynchrone Natur der OSS Redis-Replikation sowie über die verfügbaren CloudWatch Kennzahlen zur Berichterstattung über die Replikationsverzögerung zwischen Primär- und Replikatknoten. Für Anwendungsfälle, die eine höhere Datensicherheit erfordern, können Sie den OSS WAIT Redis-Befehl nutzen, um Replikate zu zwingen, Schreibvorgänge zu bestätigen, bevor sie auf verbundene Clients antworten.

[Ressourcen]:

- [Metriken für Redis OSS](#)
- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) mithilfe von Amazon CloudWatch](#)
- [Am besten] Überprüfen Sie regelmäßig die Reaktionsfähigkeit Ihrer Anwendung während eines Failovers mithilfe des ElastiCache Test-Failovers. API

[Ressourcen]:

- [Testen des automatischen Failovers auf eine Read Replica auf Amazon ElastiCache \(Redis\) OSS](#)
- [Testen des automatischen Failovers](#)

REL5: Sind Ihre ElastiCache Komponenten so konzipiert, dass sie skalierbar sind?

Einführung auf Fragenebene: Wenn Sie die Skalierungsmöglichkeiten und die verfügbaren Bereitstellungstopologien verstehen, können sich Ihre ElastiCache Komponenten im Laufe der Zeit an sich ändernde Workload-Anforderungen anpassen. ElastiCache bietet 4-Wege-Skalierung: ein-/auswärts (horizontal) sowie hoch/runter (vertikal).

Vorteil auf Frageebene: Die Einhaltung von Best Practices für ElastiCache Bereitstellungen bietet ein Höchstmaß an Flexibilität bei der Skalierung und erfüllt gleichzeitig das Well-Architected-Prinzip der horizontalen Skalierung, um die Auswirkungen von Ausfällen zu minimieren.

- [Erforderlich] Machen Sie sich mit dem Unterschied zwischen Topologien im aktivierten und im deaktivierten Clustermodus vertraut. In fast allen Fällen wird empfohlen, die Bereitstellung mit aktiviertem Clustermodus vorzunehmen, da dies im Zeitverlauf eine größere Skalierbarkeit ermöglicht. Komponenten im deaktiviertem Clustermodus können nur begrenzt horizontal skaliert werden, indem Lesereplikate hinzugefügt werden.

- [Erforderlich] Machen Sie sich damit vertraut, wann und wie Sie skalieren müssen.
 - Weitere Informationen: Fügen Sie Replikate hinzu READIOPS
 - Weitere InformationenWRITEOPS: Fügen Sie Shards hinzu (horizontal skalieren)
 - Für mehr Netzwerk-I/O – netzwerkoptimierte Instances verwenden, hochskalieren
- [Am besten] Stellen Sie Ihre ElastiCache Komponenten mit aktiviertem Cluster-Modus bereit, wobei Sie eher mehr, kleinere Knoten als weniger, größere Knoten bevorzugen. Dadurch wird der Explosionsradius eines Knotenausfalls effektiv begrenzt.
- [Am besten] Nehmen Sie Replikate in Ihre Cluster auf, um die Reaktionsfähigkeit bei Skalierungsereignissen zu verbessern
- [Gut] Verwenden Sie bei deaktiviertem Cluster-Modus Read Replicas, um die gesamte Lesekapazität zu erhöhen. ElastiCache unterstützt bis zu 5 Read Replicas im deaktivierten Cluster-Modus sowie vertikale Skalierung.
- [Ressourcen]:
 - [Skalierung von Clustern ElastiCache \(Redis\) OSS](#)
 - [Online-Hochskalieren](#)
 - [Skalierung ElastiCache für Memcached-Cluster](#)

Säule der Leistung und Effizienz von Amazon ElastiCache Well-Architected Lens

Die Säule der Leistungseffizienz konzentriert sich auf die effiziente Nutzung von IT- und Computing-Ressourcen. Zu den wichtigsten Themen gehören die Auswahl der richtigen Ressourcentypen und -größen je nach Workload-Anforderungen, die Überwachung der Leistung und das Treffen fundierter Entscheidungen, um die Effizienz aufrechtzuerhalten, wenn sich die Geschäftsanforderungen weiterentwickeln.

Themen

- [PE 1: Wie überwachen Sie die Leistung Ihres ElastiCache Amazon-Clusters?](#)
- [PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache Clusterknoten?](#)
- [PE 3: Wie verfolgen und protokollieren Sie bei Caching-Workloads die Effektivität und Leistung Ihres Caches?](#)
- [PE 4: Wie optimiert Ihr Workload den Einsatz von Netzwerkressourcen und Verbindungen?](#)
- [PE 5: Wie verwalten Sie die Löschung und/oder Bereinigung von Schlüsseln?](#)

- [PE 6: Wie modellieren und interagieren Sie mit Daten in ElastiCache?](#)
- [PE 7: Wie protokollieren Sie langsam laufende Befehle in Ihrem ElastiCache Amazon-Cluster?](#)
- [PE8: Wie hilft Auto Scaling dabei, die Leistung des ElastiCache Clusters zu steigern?](#)

PE 1: Wie überwachen Sie die Leistung Ihres ElastiCache Amazon-Clusters?

Einführung auf Fragenebene: Wenn Sie die vorhandenen Überwachungsmetriken verstehen, können Sie die aktuelle Auslastung ermitteln. Eine ordnungsgemäße Überwachung kann dazu beitragen, potenzielle Engpässe zu identifizieren, die sich auf die Leistung Ihres Clusters auswirken.

Vorteil auf Fragenebene: Wenn Sie die Metriken Ihres Clusters kennen, können Sie Optimierungsmethoden entwickeln, die zu einer geringeren Latenz und einem höheren Durchsatz führen können.

- [Erforderlich] Basisleistungstests mit einer Teilmenge Ihres Workloads.
 - Sie sollten die Leistung des tatsächlichen Workloads mithilfe von Mechanismen wie Lasttests überwachen.
 - Überwachen Sie die CloudWatch Metriken, während Sie diese Tests durchführen, um sich ein Bild von den verfügbaren Metriken zu machen und eine Leistungsbasis festzulegen.
- [Am besten] Benennen Sie für ElastiCache (Redis-OSS) Workloads rechenintensive Befehle um, z. B. um die Fähigkeit von Benutzern einzuschränken KEYS, blockierende Befehle auf Produktionsclustern auszuführen.
 - ElastiCache (Redis-OSS) Workloads, auf denen Engine 6.x ausgeführt wird, können die rollenbasierte Zugriffskontrolle nutzen, um bestimmte Befehle einzuschränken. Der Zugriff auf die Befehle kann gesteuert werden, indem Benutzer und Benutzergruppen mit der AWS Konsole erstellt oder CLI die Benutzergruppen einem (Redis-) Cluster zugeordnet werden. ElastiCache OSS In Redis OSS 6 können wir, wenn es aktiviert RBAC ist, „- @dangerous“ verwenden, wodurch teure Befehle wie KEYS, MONITORSORT, usw. für diesen Benutzer nicht zugelassen werden.
 - Benennen Sie Befehle für Engine-Version 5.x mithilfe des `rename-commands` Parameters in der Cluster-Parametergruppe ElastiCache (RedisOSS) um.
- [Besser] Analysieren Sie langsame Abfragen und suchen Sie nach Optimierungsmöglichkeiten.
 - Bei ElastiCache (Redis-OSS) Workloads erfahren Sie mehr über Ihre Abfragen, indem Sie das Slow Log analysieren. Sie können beispielsweise den Befehl `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 ElastiCache (Redis-OSS) Datenstrukturen effizienter ausgeführt werden. Für Bereichsabfragen im numerischen Stil kann eine Anwendung beispielsweise einfache numerische Indizes mit sortierten Sätzen implementieren. Die Verwaltung dieser Indizes kann die Anzahl der am Datensatz durchgeführten Scans reduzieren und Daten mit höherer Leistungseffizienz zurückgeben.
- `redis-benchmark` bietet für ElastiCache (Redis-OSS) Workloads eine einfache Oberfläche zum Testen der Leistung verschiedener Befehle mithilfe benutzerdefinierter Eingaben wie Anzahl der Clients und Datengröße.
- Da Memcached nur einfache Befehle auf Schlüsselebene unterstützt, sollten Sie erwägen, zusätzliche Schlüssel als Indizes zu erstellen, um zu vermeiden, dass der Schlüsselbereich wiederholt wird, um Client-Abfragen zu bearbeiten.
- [Ressourcen]:
 - [Überwachung der Nutzung mit Metrics CloudWatch](#)
 - [Überwachung der Nutzung mit CloudWatch Metrics](#)
 - [CloudWatch Amazon-Alarme verwenden](#)
 - [Redis-spezifische Parameter](#)
 - [SLOWLOG](#)
 - [Redis-Benchmark OSS](#)

PE 2: Wie verteilen Sie die Arbeit auf Ihre ElastiCache Clusterknoten?

Einführung auf Fragenebene: Die Art und Weise, wie Ihre Anwendung eine Verbindung zu ElastiCache Amazon-Knoten herstellt, kann sich auf die Leistung und Skalierbarkeit des Clusters auswirken.

Vorteil auf Fragenebene: Durch die richtige Nutzung der verfügbaren Knoten im Cluster wird sichergestellt, dass die Arbeit auf die verfügbaren Ressourcen verteilt wird. Die folgenden Methoden tragen ebenfalls dazu bei, ungenutzte Ressourcen zu vermeiden.

- [Erforderlich] Sorgen Sie dafür, dass sich die Clients mit dem richtigen ElastiCache Endpunkt verbinden.
 - ElastiCache (RedisOSS) implementiert je nach verwendetem Clustermodus unterschiedliche Endpunkte. Wenn der Clustermodus aktiviert ist, ElastiCache wird ein Konfigurationsendpunkt

bereitgestellt. ElastiCache Stellt bei deaktiviertem Clustermodus einen primären Endpunkt bereit, der normalerweise für Schreibvorgänge verwendet wird, und einen Leser-Endpunkt für die Verteilung von Lesevorgängen zwischen Replikaten. Die korrekte Implementierung dieser Endpunkte führt zu einer besseren Leistung und einfacheren Skalierungsvorgängen. Vermeiden Sie es, eine Verbindung zu einzelnen Knotenendpunkten herzustellen, es sei denn, dies ist ausdrücklich erforderlich.

- ElastiCache Stellt für Memcached-Cluster mit mehreren Knoten einen Konfigurationsendpunkt bereit, der Auto Discovery aktiviert. Es wird empfohlen, einen Hashing-Algorithmus zu verwenden, um die Arbeit gleichmäßig auf die Cache-Knoten zu verteilen. Viele Memcached-Clientbibliotheken implementieren konsistentes Hashing. Überprüfen Sie die Dokumentation der von Ihnen verwendeten Bibliothek darauf, ob sie konsistentes Hashing unterstützt und wie es implementiert wird. Weitere Informationen zur Implementierung dieser Funktionen finden Sie [hier](#).
- [Besser] Implementieren Sie eine Strategie zur Identifizierung und Korrektur von Tastaturbefehlen in Ihrem Workload.
 - Bedenken Sie die Auswirkungen mehrdimensionaler OSS Redis-Datenstrukturen wie Listen, Streams, Sets usw. Diese Datenstrukturen werden in einzelnen OSS Redis-Schlüsseln gespeichert, die sich auf einem einzigen Knoten befinden. Ein sehr großer multidimensionaler Schlüssel nimmt potenziell mehr Netzwerkkapazität und Speicher in Anspruch als andere Datentypen und kann eine unverhältnismäßige Nutzung dieses Knotens bewirken. Wenn möglich, sollten Sie Ihren Workload so gestalten, dass der Datenzugriff auf viele einzelne Schlüssel verteilt wird.
 - Tastaturbefehle im Workload können die Leistung des verwendeten Knotens beeinträchtigen. Bei ElastiCache (Redis-OSS) Workloads können Sie Hotkeys erkennen, `redis-cli --hotkeys` wenn eine Richtlinie zur maximalen Speicherbelegung LFU vorhanden ist.
 - Erwägen Sie, Tastaturbefehle auf mehreren Knoten zu replizieren, um den Zugriff auf sie gleichmäßiger zu verteilen. Bei diesem Ansatz muss der Client auf mehrere Primärknoten schreiben (der OSS Redis-Knoten selbst bietet diese Funktionalität nicht) und zusätzlich zum ursprünglichen Schlüsselnamen eine Liste mit Schlüsselnamen verwalten, aus denen gelesen werden kann.
 - ElastiCache([RedisOSS](#)) [Version 6 unterstützt servergestütztes clientseitiges Caching](#). Auf diese Weise können Anwendungen auf Änderungen an einem Schlüssel warten, bevor sie über das Netzwerk zurückrufen. ElastiCache
- [Ressourcen]:
 - [Konfigurieren Sie ElastiCache \(RedisOSS\) für eine höhere Verfügbarkeit](#)

- [Ermitteln von Verbindungsendpunkten](#)
- [Bewährte Methoden für den Lastausgleich](#)
- [Clientseitiges Caching in Redis OSS](#)

PE 3: Wie verfolgen und protokollieren Sie bei Caching-Workloads die Effektivität und Leistung Ihres Caches?

Einführung auf Fragenebene: Caching ist ein häufig auftretender Workload, ElastiCache und es ist wichtig, dass Sie wissen, wie Sie die Effektivität und Leistung Ihres Caches verwalten können.

Vorteil auf Fragenebene: Ihre Anwendung kann Anzeichen einer stagnierenden Leistung aufweisen. Ihre Fähigkeit, Cache-spezifische Metriken als Grundlage für Ihre Entscheidung zur Steigerung der App-Leistung zu verwenden, ist für Ihren Cache-Workload von entscheidender Bedeutung.

- [Erforderlich] Messen und verfolgen Sie die Cache-Trefferquote im Zeitverlauf. Die Effizienz Ihres Caches wird durch seine „Cache-Trefferquote“ bestimmt. Die Cache-Trefferquote wird durch die Summe der Schlüsseltreffer geteilt durch die Gesamtzahl der Treffer und Fehlversuche definiert. Je näher die Quote an 1 liegt, desto effektiver ist Ihr Cache. Eine niedrige Cache-Trefferquote wird durch die Menge der Cache-Fehlversuche verursacht. Cache-Fehlversuche treten auf, wenn der angeforderte Schlüssel nicht im Cache gefunden wird. Ein Schlüssel befindet sich nicht im Cache, weil er entweder bereinigt oder gelöscht wurde, abgelaufen ist oder nie existiert hat. Machen Sie sich klar, warum sich Schlüssel nicht im Cache befinden, und entwickeln Sie geeignete Strategien, um sie im Cache vorliegen zu haben.

[Ressourcen]:

- [Erforderlich] Messen und erfassen Sie die Leistung Ihres Anwendungs-Caches in Verbindung mit Latenz- und CPU Nutzungswerten, um zu ermitteln, ob Sie Anpassungen an Ihren time-to-live oder anderen Anwendungskomponenten vornehmen müssen. ElastiCache bietet eine Reihe von CloudWatch Metriken für aggregierte Latenzen für jede Datenstruktur. Diese Latenzmetriken werden anhand der Commandstats-Statistik des INFO Befehls ElastiCache (RedisOSS) berechnet und beinhalten nicht die Netzwerk- und I/O-Zeit. Dies ist nur die Zeit, die ElastiCache (RedisOSS) für die Verarbeitung der Operationen benötigt.

[Ressourcen]:

- [Überwachung von Best Practices mit ElastiCache \(RedisOSS\) unter Verwendung von Amazon CloudWatch](#)

- [Am besten] Wählen Sie die richtige Caching-Strategie für Ihre Anforderungen aus. Eine niedrige Cache-Trefferquote wird durch die Menge der Cache-Fehlversuche verursacht. Wenn Ihr Workload so konzipiert ist, dass es nur wenige Cache-Fehlversuche gibt (z. B. Kommunikation in Echtzeit), sollten Sie Ihre Caching-Strategien überprüfen und die für Ihren Workload am besten geeigneten Lösungen anwenden, z. B. Abfrage-Instrumente zur Messung von Speicher und Leistung. Die tatsächlichen Strategien, die Sie zum Auffüllen und Verwalten Ihres Cache implementieren müssen, hängt von den Daten, die Ihre Clients zwischenspeichern müssen, und den Zugriffsmustern auf diese Daten ab. Es ist beispielsweise unwahrscheinlich, dass Sie dieselbe Strategie sowohl für personalisierte Empfehlungen in einer Streaming-Anwendung als auch für aktuelle Nachrichten verwenden.

[Ressourcen]:

- [Caching-Strategien](#)
- [Bewährte Methoden für das Caching](#)
- [Skalierbare Leistung mit Amazon ElastiCache Whitepaper](#)

PE 4: Wie optimiert Ihr Workload den Einsatz von Netzwerkressourcen und Verbindungen?

Einführung auf Fragenebene: ElastiCache (RedisOSS) und ElastiCache (Memcached) werden von vielen Anwendungsclients unterstützt, und die Implementierungen können variieren. Sie müssen das vorhandene Netzwerk- und Verbindungsmanagement verstehen, um mögliche Auswirkungen auf die Leistung zu analysieren.

Vorteil auf Fragenebene: Die effiziente Nutzung von Netzwerkressourcen kann die Leistungseffizienz Ihres Clusters verbessern. Die folgenden Empfehlungen können die Netzwerkanforderungen reduzieren und die Latenz und den Durchsatz des Clusters verbessern.

- [Erforderlich] Verwalten Sie proaktiv Verbindungen zu Ihrem Cluster. ElastiCache
 - Verbindungspooling in der Anwendung reduziert den Overhead im Cluster, der durch das Öffnen und Schließen von Verbindungen entsteht. Überwachen Sie das Verbindungsverhalten in Amazon CloudWatch mithilfe von `CurConnections` und `NewConnections`.
 - Vermeiden Sie Verbindungslecks, indem Sie die Client-Verbindungen gegebenenfalls ordnungsgemäß schließen. Zu den Strategien zur Verbindungsverwaltung gehören das ordnungsgemäße Schließen von Verbindungen, die nicht verwendet werden, und das Festlegen von Verbindungs-Timeouts.

- Für Memcached-Workloads gibt es eine konfigurierbare Speichermenge, die für die Verarbeitung von Verbindungen reserviert ist und als `memcached_connections_overhead` bezeichnet wird.
- [Besser] Komprimieren Sie große Objekte, um den Speicherplatz zu reduzieren und den Netzwerkdurchsatz zu verbessern.
 - Datenkomprimierung kann den erforderlichen Netzwerkdurchsatz (Gbit/s) reduzieren, erhöht jedoch den Arbeitsaufwand für die Anwendung zum Komprimieren und Dekomprimieren von Daten.
 - Durch die Komprimierung wird auch der Speicherverbrauch von Schlüsseln reduziert.
 - Berücksichtigen Sie auf der Grundlage Ihrer Anwendungsanforderungen die Kompromisse zwischen Komprimierungsverhältnis und Komprimierungsgeschwindigkeit.
- [Ressourcen]:
 - [ElastiCache \(RedisOSS\) — Globaler Datenspeicher](#)
 - [Memcached-spezifische Parameter](#)
 - [ElastiCache \(RedisOSS\) 5.0.3 verbessert die I/O-Verarbeitung, um die Leistung zu steigern](#)
 - [Konfigurieren Sie ElastiCache \(RedisOSS\) für eine höhere Verfügbarkeit](#)

PE 5: Wie verwalten Sie die Löschung und/oder Bereinigung von Schlüsseln?

Einführung auf Fragenebene: Workloads haben unterschiedliche Anforderungen und erwartetes Verhalten, wenn sich ein Clusterknoten den Grenzwerten für den Speicherverbrauch nähert. ElastiCache (RedisOSS) hat unterschiedliche Richtlinien für den Umgang mit diesen Situationen.

Vorteil auf Fragenebene: Die richtige Verwaltung des verfügbaren Speichers und das Verständnis der Lösungsrichtlinien tragen dazu bei, dass das Clusterverhalten bei Überschreitung der Speicherlimits von Instances besser erkannt wird.

- [Erforderlich] Instrumentieren Sie den Datenzugriff, um zu beurteilen, welche Richtlinie angewendet werden soll. Identifizieren Sie eine geeignete Richtlinie für maximalen Arbeitsspeicher, um zu kontrollieren, ob und wie Bereinigungen im Cluster durchgeführt werden.
 - Eine Bereinigung erfolgt, wenn der maximale Arbeitsspeicher des Clusters verbraucht ist und eine Richtlinie vorhanden ist, welche die Bereinigung zulässt. Das Verhalten des Clusters in dieser Situation hängt von der angegebenen Bereinigungsrichtlinie ab. Diese Richtlinie kann mithilfe der Parametergruppe `maxmemory-policy` On the ElastiCache (RedisOSS) -Cluster verwaltet werden.

- Die Standardrichtlinie gibt Speicherplatz `volatile-lru` frei, indem Schlüssel mit einer festgelegten Ablaufzeit (Wert) entfernt werden. TTL In den Richtlinien „Am wenigsten häufig verwendet“ (LFU) und „Zuletzt verwendet“ (LRU) werden Schlüssel je nach Nutzung entfernt.
- Für Memcached-Workloads gibt es eine LRU Standardrichtlinie, die Räumungen auf jedem Knoten steuert. Die Anzahl der Räumungen in Ihrem ElastiCache Amazon-Cluster kann mithilfe der Räumungsmetrik auf Amazon überwacht werden. CloudWatch
- [Besser] Standardisieren Sie das Löschverhalten, um die Auswirkungen auf die Leistung Ihres Clusters zu kontrollieren und unerwartete Leistungsengpässe zu vermeiden.
 - Bei ElastiCache (Redis-OSS) Workloads ist das explizite Entfernen von Schlüsseln aus dem Cluster wie folgt: `DEL`: `UNLINK` Es entfernt die angegebenen Schlüssel. Der Befehl führt jedoch die eigentliche Speicherrückgewinnung in einem anderen Thread durch, sodass er nicht blockiert, während dies mit `DEL` der Fall ist. Die eigentliche Entfernung erfolgt später asynchron.
 - Für ElastiCache (RedisOSS) 6.x-Workloads kann das Verhalten des `DEL` Befehls in der Parametergruppe mithilfe von Parameter geändert werden. `lazyfree-lazy-user-del`
- [Ressourcen]:
 - [Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen](#)
 - [UNLINK](#)
 - [Cloud-Finanzmanagement mit AWS](#)

PE 6: Wie modellieren und interagieren Sie mit Daten in ElastiCache?

Einführung auf Fragenebene: ElastiCache ist stark von der Anwendung abhängig von den Datenstrukturen und dem verwendeten Datenmodell, muss aber auch der zugrunde liegende Datenspeicher (falls vorhanden) berücksichtigt werden. Machen Sie sich mit den verfügbaren ElastiCache (Redis-OSS) Datenstrukturen vertraut und stellen Sie sicher, dass Sie die für Ihre Bedürfnisse am besten geeigneten Datenstrukturen verwenden.

Vorteil auf Fragenebene: Die Datenmodellierung ElastiCache besteht aus mehreren Ebenen, einschließlich Anwendungsfällen, Datentypen und Beziehungen zwischen Datenelementen. Darüber hinaus haben alle ElastiCache (Redis-OSS) Datentypen und Befehle ihre eigenen, gut dokumentierten Leistungssignaturen.

- [Am besten] Eine bewährte Methode besteht darin, das unbeabsichtigte Überschreiben von Daten zu reduzieren. Verwenden Sie eine Namenskonvention, die überlappende Schlüsselnamen

minimiert. Bei der herkömmlichen Benennung Ihrer Datenstrukturen wird eine hierarchische Methode wie APPNAME : CONTEXT : ID bzw. ORDER-APP : CUSTOMER : 123 verwendet.

[Ressourcen]:

- [Benennung von Schlüsseln](#)
- [Best] -Befehle ElastiCache (RedisOSS) haben eine Zeitkomplexität, die durch die Big-O-Notation definiert wird. Hier ist die Komplexität eines Befehls eine algorithmische/mathematische Darstellung seiner Wirkung. Bei der Einführung eines neuen Datentyps in Ihre Anwendung müssen Sie die Zeitkomplexität der zugehörigen Befehle sorgfältig überprüfen. Befehle mit einer Zeitkomplexität von $O(1)$ sind zeitlich konstant und hängen nicht von der Größe der Eingabe ab. Befehle mit einer Zeitkomplexität von $O(N)$ sind jedoch zeitlich linear und hängen von der Größe der Eingabe ab. Aufgrund des Single-Thread-Designs von ElastiCache (RedisOSS) führt ein großes Volumen an Vorgängen mit hoher Zeitkomplexität zu einer geringeren Leistung und möglichen Zeitüberschreitungen bei Operationen.

[Ressourcen]:

- [Befehle](#)
- [Am besten] Verwenden Sie diese APIs Option, um sich einen GUI Überblick über das Datenmodell in Ihrem Cluster zu verschaffen.

[Ressourcen]:

- [Redis OSS Commander](#)
- [Redis-Browser OSS](#)
- [Redsmin](#)

PE 7: Wie protokollieren Sie langsam laufende Befehle in Ihrem ElastiCache Amazon-Cluster?

Einführung auf Fragenebene: Die Leistungsoptimierung bietet Vorteile durch die Erfassung, Aggregation und Benachrichtigung von Befehlen mit langer Laufzeit. Wenn Sie wissen, wie lange die Ausführung von Befehlen dauert, können Sie feststellen, welche Befehle zu einer schlechten Leistung führen und welche Befehle die Engine daran hindern, optimal zu funktionieren. ElastiCache (RedisOSS) kann diese Informationen auch an Amazon CloudWatch oder Amazon Kinesis Data Firehose weiterleiten.

Vorteil auf Fragenebene: Die Protokollierung an einem festen Standort und die Bereitstellung von Benachrichtigungsereignissen bei langsamen Befehlen können zu einer detaillierten Leistungsanalyse beitragen und zur Auslösung automatisierter Ereignisse verwendet werden.

- [Erforderlich] Amazon ElastiCache (RedisOSS) führt Engine Version 6.0 oder neuer aus, ordnungsgemäß konfigurierte Parametergruppe und SLOWLOG Protokollierung sind auf dem Cluster aktiviert.
 - Die erforderlichen Parameter sind nur verfügbar, wenn die Engine-Versionskompatibilität auf OSS Redis-Version 6.0 oder höher eingestellt ist.
 - SLOWLOGDie Protokollierung erfolgt, wenn die Serverausführungszeit eines Befehls länger dauert als ein bestimmter Wert. Das Verhalten des Clusters hängt von den zugehörigen Parametergruppenparametern ab, d. h. `slowlog-log-slower-than` und `slowlog-max-len`.
 - Änderungen werden sofort wirksam.
- [Am besten] Nutzen Sie unsere Kinesis Data Firehose Firehose-Funktionen. CloudWatch
 - Nutzen Sie die Filter- und Alarmfunktionen von CloudWatch CloudWatch Logs Insights und Amazon Simple Notification Services, um die Leistung zu überwachen und Ereignisse zu benachrichtigen.
 - Verwenden Sie die Streaming-Funktionen von Kinesis Data Firehose, um SLOWLOG Protokolle im permanenten Speicher zu archivieren oder um eine automatische Cluster-Parameteroptimierung auszulösen.
 - Finden Sie heraus, JSON ob TEXT das einfache Format Ihren Anforderungen am besten entspricht.
 - Erteilen Sie IAM Berechtigungen zum Veröffentlichen in CloudWatch oder Kinesis Data Firehose.
- [Besser] Konfigurieren Sie `slowlog-log-slower-than` auf einen anderen Wert als den Standardwert.
 - Dieser Parameter bestimmt, wie lange ein Befehl innerhalb der OSS Redis-Engine ausgeführt werden kann, bevor er als langsam laufender Befehl protokolliert wird. Der Standardwert ist 10 000 Mikrosekunden (10 Millisekunden). Der Standardwert ist für einige Workloads möglicherweise zu hoch.
 - Ermitteln Sie auf der Grundlage der Anwendungsanforderungen und der Testergebnisse einen Wert, der für Ihren Workload besser geeignet ist. Ein zu niedriger Wert kann jedoch zu übermäßigen Daten führen.
- [Besser] Übernehmen Sie für `slowlog-max-len` den Standardwert.

- Dieser Parameter bestimmt die Obergrenze dafür, wie viele langsam laufende Befehle zu einem bestimmten Zeitpunkt im OSS Redis-Speicher erfasst werden. Mit dem Wert 0 wird die Erfassung effektiv deaktiviert. Je höher der Wert, desto mehr Einträge werden im Speicher gespeichert. Dadurch wird die Wahrscheinlichkeit verringert, dass wichtige Informationen bereinigt werden, bevor sie überprüft werden können. Der Standardwert lautet 128.
- Der Standardwert ist für die meisten Workloads geeignet. Wenn es notwendig ist, Daten in einem erweiterten Zeitfenster von der Redis-CLI aus über den SLOWLOG Befehl zu analysieren, sollten Sie erwägen, diesen Wert zu erhöhen. Dadurch können mehr Befehle im Redis-Speicher verbleiben. OSS

Wenn Sie die SLOWLOG Daten entweder an CloudWatch Logs oder Kinesis Data Firehose weitergeben, werden die Daten dauerhaft gespeichert und können außerhalb des ElastiCache Systems analysiert werden, wodurch die Notwendigkeit reduziert wird, eine große Anzahl langsam laufender Befehle im Redis-Speicher zu speichern. OSS

- [Ressourcen]:
 - [Wie aktiviere ich Redis OSS Slow Log in einem ElastiCache \(Redis-\) Cache-Cluster? OSS](#)
 - [Protokollzustellung](#)
 - [Redis-spezifische Parameter OSS](#)
 - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
 - [Amazon Kinesis Data Firehose](#)

PE8: Wie hilft Auto Scaling dabei, die Leistung des ElastiCache Clusters zu steigern?

Einführung auf Fragenebene: Durch die Implementierung der Funktion von Redis OSS Auto Scaling können sich Ihre ElastiCache Komponenten im Laufe der Zeit anpassen, um die gewünschten Shards oder Replicas automatisch zu erhöhen oder zu verringern. Dies kann erreicht werden, indem entweder die Zielverfolgung oder die geplante Skalierungsrichtlinie implementiert wird.

Vorteil auf Fragenebene: Wenn Sie die Arbeitslastspitzen verstehen und entsprechend planen, können Sie eine verbesserte Caching-Leistung und einen unterbrechungsfreien Betrieb sicherstellen. ElastiCache (RedisOSS) Auto Scaling überwacht kontinuierlich Ihre CPU /Memory-Auslastung, um sicherzustellen, dass Ihr Cluster mit den gewünschten Leistungsniveaus arbeitet.

- [Erforderlich] Beim Starten eines Clusters für ElastiCache (OSSRedis):
 1. Stellen Sie sicher, dass der Clustermodus aktiviert ist.

2. Stellen Sie sicher, dass die Instance zu einer Familie mit einem bestimmten Typ und einer bestimmten Größe gehört, die Auto Scaling unterstützt.
3. Vergewissern Sie sich, dass der Cluster nicht in globalen Datenspeichern, Outposts oder lokalen Zonen ausgeführt wird

[Ressourcen]:

- [Skalierung von Clustern in Redis OSS \(Clustermodus aktiviert\)](#)
- [Verwenden von Auto Scaling mit Shards](#)
- [Verwenden von Auto Scaling mit Replikaten](#)
- [Am besten] Stellen Sie fest, ob Ihr Workload lese- oder schreibintensiv ist, um die Skalierungsrichtlinie zu definieren. Verwenden Sie nur eine Nachverfolgungsmetrik, um eine optimale Leistung zu erzielen. Es wird empfohlen, für jede Dimension mehrere Richtlinien zu vermeiden, da Auto-Scaling-Richtlinien aufskalieren, wenn das Ziel erreicht wird, aber erst dann abskalieren, wenn alle Ziel-Nachverfolgungsrichtlinien für die Abskalierung bereit sind.

[Ressourcen]:

- [Auto-Scaling-Richtlinien](#)
- [Definieren einer Skalierungsrichtlinie](#)
- [Am besten] Wenn Sie die Leistung im Zeitverlauf überwachen, können Sie Workload-Änderungen erkennen, die bei punktueller Überwachung unbemerkt bleiben würden. Sie können die entsprechenden CloudWatch Metriken für die Clusterauslastung über einen Zeitraum von vier Wochen analysieren, um den Zielschwellenwert zu bestimmen. Wenn Sie sich immer noch nicht sicher sind, welchen Wert Sie wählen möchten, empfehlen wir, mit dem minimal unterstützten vordefinierten Metrikwert zu beginnen.

[Ressourcen]:

- [Überwachung der Nutzung mit Metriken CloudWatch](#)
- [Besser] Wir empfehlen, Ihre Anwendung mit den erwarteten minimalen und maximalen Workloads zu testen, um die genaue Anzahl der Shard/Replikate zu ermitteln, die der Cluster benötigt, um Skalierungsrichtlinien zu entwickeln und Verfügbarkeitsprobleme zu minimieren.

[Ressourcen]:

- [Registrieren eines skalierbaren Ziels](#)
- [Registrieren eines skalierbaren Ziels](#)

Säule zur ElastiCache Kostenoptimierung bei Amazon Well-Architected Lens

Die Säule der Kostenoptimierung konzentriert sich auf die Vermeidung unnötiger Kosten. Zu den wichtigsten Themen gehören das Verständnis und die Kontrolle, wofür Geld ausgegeben wird, die Auswahl des am besten geeigneten Knotentyps (verwenden Sie Instances, die das Daten-Tiering auf der Grundlage der Workload-Anforderungen unterstützen), die richtige Anzahl von Ressourcentypen (wie viele Lesereplikate), die Analyse der Ausgaben im Zeitverlauf und die Skalierung, um Geschäftsanforderungen zu erfüllen, ohne zu viel Geld auszugeben.

Themen

- [COST1: Wie identifizieren und verfolgen Sie die mit Ihren ElastiCache Ressourcen verbundenen Kosten? Wie entwickeln Sie Mechanismen, die es Benutzern ermöglichen, Ressourcen zu erstellen, zu verwalten und zu entsorgen?](#)
- [COST2: Wie nutzen Sie Tools zur kontinuierlichen Überwachung, um die mit Ihren ElastiCache Ressourcen verbundenen Kosten zu optimieren?](#)
- [COST3: Sollten Sie einen Instance-Typ verwenden, der Data Tiering unterstützt? Was sind die Vorteile von Daten-Tiering? Wann sollten Daten-Tiering-Instances nicht verwendet werden?](#)

COST1: Wie identifizieren und verfolgen Sie die mit Ihren ElastiCache Ressourcen verbundenen Kosten? Wie entwickeln Sie Mechanismen, die es Benutzern ermöglichen, Ressourcen zu erstellen, zu verwalten und zu entsorgen?

Einführung auf Fragenebene: Um Kostenmetriken zu verstehen, müssen mehrere Teams beteiligt werden und zusammenarbeiten: Softwareentwicklung, Datenmanagement, Produktbesitzer, Finanzen und Führung. Um die wichtigsten Kostentreiber zu identifizieren, müssen alle Beteiligten die Hebel der Kontrolle der Servicenutzung und die Kompromisse beim Kostenmanagement verstehen. Dies ist häufig der entscheidende Unterschied zwischen erfolgreichen und weniger erfolgreichen Bemühungen zur Kostenoptimierung. Wenn Sie sicherstellen, dass Sie über Prozesse und Tools verfügen, mit denen Sie die Ressourcen verfolgen können, die von der Entwicklung über die Produktion bis hin zur Außerbetriebnahme entstanden sind, können Sie die damit verbundenen Kosten besser kontrollieren.

Vorteil auf Fragenebene: Die kontinuierliche Erfassung aller mit Ihrem Workload verbundenen Kosten erfordert ein tiefes Verständnis der Architektur, zu der ElastiCache auch eine ihrer Komponenten

gehört. Darüber hinaus sollten Sie über einen Kostenmanagementplan verfügen, um die Nutzung zu erfassen und mit Ihrem Budget zu vergleichen.

- [Erforderlich] Richten Sie ein Cloud Center of Excellence (CCoE) mit einer der Gründungschartas ein, das die Definition, Nachverfolgung und Umsetzung von Kennzahlen rund um die Nutzung in Ihrem Unternehmen selbst übernimmt. ElastiCache Falls eines CCoE existiert und funktioniert, stellen Sie sicher, dass es weiß, wie die damit verbundenen Kosten zu lesen und nachzuverfolgen sind. ElastiCache Wenn Ressourcen erstellt werden, überprüfen Sie anhand von IAM Rollen und Richtlinien, ob nur bestimmte Teams und Gruppen Ressourcen instanzieren können. Damit wird sichergestellt, dass die Kosten mit den Geschäftsergebnissen in Verbindung stehen und dass im Hinblick auf die Kosten eine klare Linienverantwortung festgelegt wird.
 1. CCoE sollte Kostenkennzahlen identifizieren, definieren und veröffentlichen, die regelmäßig — monatlich — aktualisiert werden und sich auf die wichtigsten ElastiCache Nutzungsdaten beziehen, z. B.:
 - a. Verwendete Knotentypen und ihre Attribute: Standard oder speicheroptimiert, On-Demand-Instances oder Reserved Instances, Regionen und Availability Zones
 - b. Arten von Umgebungen: kostenlose Umgebungen, Entwicklungsumgebungen, Testumgebungen und Produktionsumgebungen
 - c. Strategien zur Speicherung und Aufbewahrung von Backups
 - d. Datentransfer innerhalb und zwischen Regionen
 - e. Instances, die in Amazon Outposts ausgeführt werden
 2. CCoE besteht aus einem funktionsübergreifenden Team, in dem nicht ausschließlich die Teams für Softwareentwicklung, Datenmanagement, Produktteam, Finanzen und Führungskräfte in Ihrem Unternehmen vertreten sind.

[Ressourcen]:

- [Schaffen eines Cloud-Kompetenzzentrums](#)
- [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Verwenden Sie Kostenzuordnungs-Tags, um die Kosten auf einer niedrigen Granularitätsstufe zu verfolgen. Verwenden Sie AWS Cost Management, um Ihre AWS Kosten und Nutzung im Laufe der Zeit zu visualisieren, zu verstehen und zu verwalten.
 1. Verwenden Sie Tags, um Ihre Ressourcen zu organisieren, und Tags zur Kostenzuweisung, um Ihre AWS Kosten detailliert zu verfolgen. Nachdem Sie die Kostenzuordnungs-Tags aktiviert haben, AWS verwendet es die Kostenzuordnungs-Tags, um Ihre Ressourcenkosten in Ihrem Kostenzuordnungsbericht zu organisieren, sodass Sie Ihre AWS Kosten leichter kategorisieren

und verfolgen können. AWS stellt zwei Arten von Kostenverrechnungs-Tags bereit: AWS generierte Tags und benutzerdefinierte Tags. AWS definiert, erstellt und wendet die AWS generierten Tags für Sie an, und Sie definieren, erstellen und wenden benutzerdefinierte Tags an. Sie müssen beide Arten von Tags separat aktivieren, bevor sie in AWS-Kostenmanagement oder einem Kostenzuordnungsbericht angezeigt werden können.

2. Verwenden Sie Tags für die Kostenzuweisung, um Ihre AWS Rechnung so zu organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Wenn Sie Ihren Ressourcen in Amazon Kostenzuordnungs-Tags hinzufügen ElastiCache, können Sie die Kosten verfolgen, indem Sie die Ausgaben auf Ihren Rechnungen nach Ressourcen-Tag-Werten gruppieren. Erwägen Sie, ob Sie Tags miteinander kombinieren, um Kosten detaillierter zu verfolgen.

[Ressourcen]:

- [Verwenden von AWS Kostenzuweisungs-Tags](#)
 - [Überwachen von Kosten mit Kostenzuordnungs-Tags](#)
 - [AWS Cost Explorer](#)
- [Am besten] Verknüpfen Sie ElastiCache Kosten mit Kennzahlen, die sich auf das gesamte Unternehmen auswirken.
 1. Berücksichtigen Sie sowohl Geschäftsmetriken als auch betriebliche Metriken wie die Latenz – welche Konzepte in Ihrem Geschäftsmodell sind rollenübergreifend verständlich? Die Metriken müssen für so viele Rollen wie möglich in der Organisation verständlich sein.
 2. Beispiele: gleichzeitig bereitgestellte Benutzer, maximale und durchschnittliche Latenz pro Operation und Benutzer, Benutzerbindungsergebnisse, Rückkehrraten von Benutzern pro Woche, Sitzungslänge/Benutzer, Abbruchrate, Cache-Trefferrate und verfolgte Schlüssel

[Ressourcen]:

- [Überwachung der Nutzung mit CloudWatch Metriken](#)
- [Gut] Sorgen Sie für einen Überblick über die up-to-date Architektur und den Betrieb der Kennzahlen und Kosten für den gesamten verwendeten Workload ElastiCache.
 1. Verstehen Sie Ihr gesamtes Lösungssystem, das in der ElastiCache Regel Teil eines vollständigen Ökosystems von AWS Diensten in seinem Technologieangebot ist, von Kunden über API Gateway und QuickSight Redshift bis hin zu Reporting-Tools (zum Beispiel).
 2. Tragen Sie die Komponenten Ihrer Lösung wie Clients, Verbindungen, Sicherheit, In-Memory-Betrieb, Speicher, Ressourcenautomatisierung, Datenzugriff und -verwaltung in Ihrem Architekturdiagramm auf. Jede Ebene ist mit der gesamten Lösung verbunden und hat eigene

Anforderungen und Funktionen, welche die Gesamtkosten erhöhen und/oder Ihnen helfen, diese zu verwalten.

3. Ihr Diagramm sollte die Nutzung von Rechenleistung, Netzwerk, Speicher, Lebenszyklusrichtlinien, die Erfassung von Kennzahlen sowie die betrieblichen und funktionalen ElastiCache Elemente Ihrer Anwendung beinhalten
4. Die Anforderungen Ihres Workloads werden sich wahrscheinlich im Laufe der Zeit ändern. Es ist wichtig, dass Sie Ihr Wissen über die zugrundeliegenden Komponenten sowie Ihre primären funktionalen Ziele aufrechterhalten und dokumentieren, um Ihr Workload-Kostenmanagement proaktiv zu gestalten.
5. Die Unterstützung der Geschäftsleitung in Bezug auf Transparenz, Rechenschaftspflicht, Priorisierung und Ressourcen ist entscheidend für eine effektive Kostenmanagementstrategie für Ihr Unternehmen. ElastiCache

COST2: Wie nutzen Sie Tools zur kontinuierlichen Überwachung, um die mit Ihren ElastiCache Ressourcen verbundenen Kosten zu optimieren?

Einführung auf Fragenebene: Sie müssen ein ausgewogenes Verhältnis zwischen Ihren ElastiCache Kosten- und Anwendungsleistungskennzahlen anstreben. Amazon CloudWatch bietet Einblicke in wichtige Betriebskennzahlen, anhand derer Sie beurteilen können, ob Ihre ElastiCache Ressourcen im Verhältnis zu Ihren Anforderungen über- oder unterausgelastet sind. Im Hinblick auf die Kostenoptimierung müssen Sie verstehen, wann Ihre Ressourcen überlastet sind, und in der Lage sein, geeignete Mechanismen zu entwickeln, um die Größe Ihrer ElastiCache Ressourcen zu ändern und gleichzeitig Ihre Betriebs-, Verfügbarkeits-, Belastbarkeits- und Leistungsanforderungen aufrechtzuerhalten.

Vorteil auf Fragenebene: Im Idealfall haben Sie genügend Ressourcen bereitgestellt, um die betrieblichen Anforderungen Ihres Workloads zu erfüllen, und verfügen nicht über unterausgelastete Ressourcen, was zu einer suboptimalen Kostensituation führen kann. Sie müssen in der Lage sein, überdimensionierte ElastiCache Ressourcen sowohl zu identifizieren als auch zu vermeiden, sie über einen längeren Zeitraum zu betreiben.

- [Erforderlich] Wird verwendet CloudWatch , um Ihre ElastiCache Cluster zu überwachen und zu analysieren, wie sich diese Metriken auf Ihre AWS Cost Explorer Explorer-Dashboards beziehen.
 1. ElastiCache bietet sowohl Metriken auf Host-Ebene (z. B. CPU Nutzung) als auch Metriken, die für die Cache-Engine-Software spezifisch sind (z. B. Cache-Abrufe und Cache-Fehlschläge).

Diese Metriken werden für jeden Cache-Knoten in 60-Sekunden-Intervallen erfasst und veröffentlicht.

2. ElastiCache Leistungskennzahlen (CPUUtilization,, EngineUtilizationSwapUsage, und Räumungen) können darauf hinweisen CurrConnections, dass Sie nach oben oder unten (mit größeren/kleineren Cache-Knotentypen) oder nach innen oder außen (mehr/weniger Shards hinzufügen) skalieren müssen. Machen Sie sich mit den Auswirkungen von Skalierungsentscheidungen auf die Kosten vertraut, indem Sie eine Playbook-Matrix erstellen, in der die zusätzlichen Kosten sowie die Mindest- und Höchstdauer geschätzt werden, die erforderlich sind, um die Schwellenwerte für die Anwendungsleistung zu erreichen.

[Ressourcen]:

- [Überwachung der Nutzung mithilfe von Metriken CloudWatch](#)
 - [Welche Metriken sollte ich überwachen?](#)
 - [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Verstehen und dokumentieren Sie Ihre Backup-Strategie und die Auswirkungen auf die Kosten.
 1. Mit ElastiCache, die Backups werden in Amazon S3 gespeichert, was eine dauerhafte Speicherung bietet. Sie müssen die Kostenauswirkungen verstehen, die sich aus Ihrer Fähigkeit ergeben, sich nach Ausfällen zu erholen.
 2. Aktivieren Sie automatische Backups, mit denen Backup-Dateien gelöscht werden, die das Aufbewahrungslimit überschritten haben.

[Ressourcen]:

- [Planen automatischer Backups](#)
 - [Amazon Simple Storage Service – Preise](#)
- [Am besten] Verwenden Sie reservierte Knoten für Ihre Instances als bewusste Strategie zur Kostenkontrolle für Workloads, die wohlverstanden und dokumentiert sind. Für reservierte Knoten wird eine Vorausgebühr erhoben, die von der Art des Knotens und der Dauer der Reservierung – ein oder drei Jahre – abhängt. Diese Gebühr ist wesentlich geringer als die stündliche Nutzungsgebühr, die bei On-Demand-Knoten anfällt.
 1. Möglicherweise müssen Sie Ihre ElastiCache Cluster mit On-Demand-Knoten betreiben, bis Sie genügend Daten gesammelt haben, um die Reserved Instance-Anforderungen abzuschätzen. Planen und dokumentieren Sie die Ressourcen, die Sie benötigen, um Ihre Anforderungen zu erfüllen, und vergleichen Sie die erwarteten Kosten für die verschiedenen Instance-Typen ([On-Demand-Instances und Reserved Instances](#)).

2. Evaluieren Sie regelmäßig die neuen verfügbaren Cache-Knotentypen und beurteilen Sie, ob es aus Sicht der Kosten- und Betriebsmetriken sinnvoll ist, Ihre Instance-Flotte auf neue Cache-Knotentypen zu migrieren.

COST3: Sollten Sie einen Instance-Typ verwenden, der Data Tiering unterstützt?

Was sind die Vorteile von Daten-Tiering? Wann sollten Daten-Tiering-Instances nicht verwendet werden?

Einführung auf Fragenebene: Die Auswahl des geeigneten Instance-Typs wirkt sich nicht nur auf die Leistung und das Service-Level aus, sondern hat auch finanzielle Auswirkungen. Mit Instance-Typen sind unterschiedliche Kosten verbunden. Die Auswahl eines oder mehrerer großer Instance-Typen, die alle Speicheranforderungen im Arbeitsspeicher erfüllen, könnte sich anbieten. Dies könnte jedoch erhebliche Kostenauswirkungen haben, sobald das Projekt ausgereift ist. Um sicherzustellen, dass der richtige Instance-Typ ausgewählt ist, muss die Leerlaufzeit von ElastiCache Objekten regelmäßig überprüft werden.

Vorteil auf Fragenebene: Sie sollten genau wissen, wie sich verschiedene Instance-Typen derzeit und in Zukunft auf Ihre Kosten auswirken. Geringe oder regelmäßige Workload-Änderungen sollten nicht zu unverhältnismäßigen Kostenänderungen führen. Wenn der Workload es zulässt, bieten Instance-Typen, die Daten-Tiering unterstützen, einen besseren Preis pro verfügbarem Speicher. Aufgrund der pro Instanz verfügbaren SSD Speicherdaten unterstützen Tiering-Instances eine wesentlich höhere Gesamtdatenmenge pro Instanz.

- [Erforderlich] Kennen der Einschränkungen von Daten-Tiering-Instances
 1. Nur für ElastiCache (Redis-OSS) Cluster verfügbar.
 2. Nur begrenzte Instance-Typen unterstützen Daten-Tiering.
 3. Nur ElastiCache (RedisOSS) Version 6.2 und höher wird unterstützt
 4. Große Artikel werden nicht ausgetauscht. SSD Objekte über 128 MiB werden im Speicher aufbewahrt.

[Ressourcen]:

- [Daten-Tiering](#)
- [ElastiCacheAmazon-Preisgestaltung](#)
- [Erforderlich] Finden Sie heraus, auf welchen Anteil Ihrer Datenbank in Prozent Ihr Workload regelmäßig zugreift.

1. Daten-Tiering-Instances eignen sich für Workloads, die häufig auf einen kleinen Teil Ihres gesamten Datensatzes zugreifen, aber dennoch einen schnellen Zugriff auf die verbleibenden Daten benötigen. Mit anderen Worten, das Verhältnis von sehr häufig zu häufig abgerufenen Daten beträgt etwa 20:80.
 2. Entwickeln Sie eine Nachverfolgung der Leerlaufzeit von Objekten auf Clusterebene.
 3. Große Implementierungen mit über 500 GB Daten sind gute Kandidaten.
- [Erforderlich] Machen Sie sich bewusst, dass Daten-Tiering-Instances für bestimmte Workloads nicht optional sind.
 1. Der Zugriff auf weniger häufig verwendete Objekte verursacht geringe Leistungseinbußen, da diese auf lokale SSD Objekte ausgelagert werden. Wenn Ihre Anwendung empfindlich auf die Reaktionszeit reagiert, testen Sie die Auswirkungen auf Ihren Workload.
 2. Nicht geeignet für Caches, die hauptsächlich große Objekte mit einer Größe von über 128 MiB speichern.

[Ressourcen]:

- [Einschränkungen](#)
- [Am besten] Reserved-Instance-Typen unterstützen Daten-Tiering. Dadurch werden die niedrigsten Kosten in Bezug auf die Menge an Datenspeicher pro Instance gewährleistet.
 1. Möglicherweise müssen Sie Ihre ElastiCache Cluster mit Instances ohne Datenklassifizierung betreiben, bis Sie Ihre Anforderungen besser verstanden haben.
 2. Analysieren Sie das ElastiCache Datennutzungsmuster Ihres Clusters.
 3. Erstellen Sie einen automatisierten Auftrag, der in regelmäßigen Abständen die Leerlaufzeit von Objekten erfasst.
 4. Wenn Sie feststellen, dass ein großer Prozentsatz (etwa 80 %) der Objekte für einen Zeitraum inaktiv ist, der für Ihren Workload als angemessen erachtet wird, dokumentieren Sie die Ergebnisse und schlagen Sie vor, den Cluster auf Instances zu migrieren, die Daten-Tiering unterstützen.
 5. Evaluieren Sie regelmäßig die neuen verfügbaren Cache-Knotentypen und beurteilen Sie, ob es aus Sicht der Kosten- und Betriebsmetriken sinnvoll ist, Ihre Instance-Flotte auf neue Cache-Knotentypen zu migrieren.

[Ressourcen]:

- [OBJECT IDLETIME](#)
- [ElastiCacheAmazon-Preisgestaltung](#)

Allgemeine Schritte und bewährte Methoden zur Fehlerbehebung

Themen

- [Verbindungsprobleme](#)
- [Fehler beim Redis OSS-Client](#)
- [Fehlerbehebung bei hoher Latenz in Serverless ElastiCache](#)
- [Behebung von Drosselungsproblemen in Serverless ElastiCache](#)
- [Verwandte Themen](#)

Verbindungsprobleme

Wenn Sie keine Verbindung zu Ihrem ElastiCache Cache herstellen können, sollten Sie eine der folgenden Möglichkeiten in Betracht ziehen:

1. Verwenden von TLS: Wenn beim Versuch, eine Verbindung zu Ihrem ElastiCache Endpunkt herzustellen, eine Verbindung nicht mehr hergestellt wird, verwenden Sie TLS möglicherweise nicht in Ihrem Client. Wenn Sie ElastiCache Serverless verwenden, ist die Verschlüsselung bei der Übertragung immer aktiviert. Stellen Sie sicher, dass Ihr Client TLS verwendet, um eine Verbindung zum Cache herzustellen. Weitere Informationen zum Herstellen einer Verbindung zu einem TLS-fähigen Cache [finden Sie hier](#).
2. VPC: Auf ElastiCache Caches kann nur innerhalb einer VPC zugegriffen werden. Stellen Sie sicher, dass die EC2-Instance, von der aus Sie auf den Cache zugreifen, und der ElastiCache Cache in derselben VPC erstellt wurden. Alternativ müssen Sie das [VPC-Peering zwischen der VPC](#), in der sich Ihre EC2-Instance befindet, und der VPC, in der Sie Ihren Cache erstellen, aktivieren.
3. Sicherheitsgruppen: ElastiCache verwendet Sicherheitsgruppen, um den Zugriff auf Ihren Cache zu kontrollieren. Berücksichtigen Sie dabei Folgendes:
 - a. Stellen Sie sicher, dass die von Ihrem ElastiCache Cache verwendete Sicherheitsgruppe eingehenden Zugriff von Ihrer EC2-Instance aus darauf ermöglicht. [Hier](#) erfahren Sie, wie Sie Regeln für eingehenden Datenverkehr in Ihrer Sicherheitsgruppe korrekt einrichten.
 - b. Stellen Sie sicher, dass die von Ihrem ElastiCache Cache verwendete Sicherheitsgruppe den Zugriff auf die Ports Ihres Caches ermöglicht (6379 und 6380 für serverlose Verbindungen und standardmäßig 6379 für selbst entworfene Ports). ElastiCache verwendet diese Ports, um Redis OSS-Befehle zu akzeptieren. Erfahren Sie [hier](#) mehr darüber, wie Sie den Portzugriff einrichten.

Fehler beim Redis OSS-Client

ElastiCache Serverless ist nur über Redis OSS-Clients zugänglich, die das Redis OSS-Clustermodus-Protokoll unterstützen. Auf selbst entworfene Cluster kann von Redis OSS-Clients in beiden Modi zugegriffen werden, abhängig von der Clusterkonfiguration.

Wenn bei Ihrem Client Redis OSS-Fehler auftreten, sollten Sie Folgendes beachten:

1. Clustermodus: Wenn bei Ihnen CROSSLOT-Fehler oder Fehler mit dem Befehl [SELECT](#) Redis OSS auftreten, versuchen Sie möglicherweise, mit einem Redis OSS-Client, der das Redis OSS-Cluster-Protokoll nicht unterstützt, auf einen Cache mit aktiviertem Clustermodus zuzugreifen. ElastiCache Serverless unterstützt nur Redis OSS-Clients, die das Redis OSS-Clusterprotokoll unterstützen. Wenn Sie Redis OSS im Modus „Cluster Mode Disabled“ (CMD) verwenden möchten, müssen Sie Ihren eigenen Cluster entwerfen.
2. CROSSLOT-Fehler: Wenn der `ERR CROSSLOT Keys in request don't hash to the same slot` Fehler auftritt, versuchen Sie möglicherweise, auf Schlüssel zuzugreifen, die nicht zu demselben Steckplatz in einem Cache im Clustermodus gehören. Zur Erinnerung: ElastiCache Serverless arbeitet immer im Clustermodus. Operationen mit mehreren Schlüsseln, Transaktionen oder Lua-Skripten mit mehreren Schlüsseln sind nur zulässig, wenn sich alle beteiligten Schlüssel im selben Hash-Slot befinden.

[Weitere bewährte Methoden zur Konfiguration von Redis OSS-Clients finden Sie in diesem Blogbeitrag.](#)

Fehlerbehebung bei hoher Latenz in Serverless ElastiCache

Wenn bei Ihrer Workload eine hohe Latenz auftritt, können Sie anhand der `SuccessfulWriteRequestLatency` Messwerte CloudWatch `SuccessfulReadRequestLatency` und überprüfen, ob die Latenz mit ElastiCache Serverless zusammenhängt. Diese Metriken messen die Latenz, die innerhalb von ElastiCache Serverless liegt. Die clientseitige Latenz und die Netzwerkausfallzeiten zwischen Ihrem Client und dem ElastiCache serverlosen Endpunkt sind nicht enthalten.

Fehlerbehebung bei der clientseitigen Latenz

Wenn Sie eine erhöhte Latenz auf der Clientseite, aber keinen entsprechenden Anstieg `CloudWatch SuccessfulReadRequestLatency` der `SuccessfulWriteRequestLatency` Messwerte für die serverseitige Latenz feststellen, sollten Sie Folgendes beachten:

- Stellen Sie sicher, dass die Sicherheitsgruppe den Zugriff auf die Ports 6379 und 6380 zulässt: ElastiCache Serverless verwendet den 6379-Port für den primären Endpunkt und den 6380-Port für den Leser-Endpunkt. Einige Clients stellen für jede neue Verbindung eine Verbindung zu beiden Ports her, auch wenn Ihre Anwendung die Funktion „Aus Replikat lesen“ nicht verwendet. Wenn Ihre Sicherheitsgruppe keinen eingehenden Zugriff auf beide Ports zulässt, kann der Verbindungsaufbau länger dauern. Erfahren Sie [hier](#) mehr darüber, wie Sie den Portzugriff einrichten.

Fehlerbehebung bei serverseitiger Latenz

Einige Schwankungen und gelegentliche Spitzenwerte sollten keinen Anlass zur Sorge geben. Wenn die Average Statistik jedoch einen starken Anstieg zeigt und anhält, sollten Sie im AWS Health Dashboard und in Ihrem Personal Health Dashboard nach weiteren Informationen suchen. Falls erforderlich, erwägen Sie, einen Support-Fall mit zu eröffnen. AWS Support

Ziehen Sie die folgenden bewährten Methoden und Strategien zur Reduzierung der Latenz in Betracht:

- **Read from Replica aktivieren:** Wenn Ihre Anwendung dies zulässt, empfehlen wir, die Funktion „Read from Replica“ in Ihrem Redis OSS-Client zu aktivieren, um Lesevorgänge zu skalieren und eine geringere Latenz zu erreichen. Wenn diese Option aktiviert ist, versucht ElastiCache Serverless, Ihre Leseanfragen an Replikat-Cache-Knoten weiterzuleiten, die sich in derselben Availability Zone (AZ) wie Ihr Client befinden, wodurch AZ-übergreifende Netzwerklatenz vermieden wird. Beachten Sie, dass die Aktivierung der Funktion „Aus Replikat lesen“ in Ihrem Client bedeutet, dass Ihre Anwendung eine eventuelle Datenkonsistenz akzeptiert. Ihre Anwendung empfängt möglicherweise für einige Zeit ältere Daten, wenn Sie versuchen, sie zu lesen, nachdem Sie in einen Schlüssel geschrieben haben.
- **Stellen Sie sicher, dass Ihre Anwendung in denselben AZs wie Ihr Cache bereitgestellt wird:** Möglicherweise stellen Sie eine höhere clientseitige Latenz fest, wenn Ihre Anwendung nicht in denselben AZs wie Ihr Cache bereitgestellt wird. Wenn Sie einen serverlosen Cache erstellen, können Sie die Subnetze angeben, von denen aus Ihre Anwendung auf den Cache zugreift, und ElastiCache Serverless erstellt VPC-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 Serverlose Anfragen werden über eine TLS-fähige TCP-Verbindung unter Verwendung des RESP-Protokolls gestellt. Das Initiieren der Verbindung

(einschließlich der Authentifizierung der Verbindung, falls konfiguriert) nimmt Zeit in Anspruch, sodass die Latenz der ersten Anfrage höher als üblich ist. Anfragen über eine bereits initialisierte Verbindung sorgen für eine gleichbleibend niedrige ElastiCache Latenz. Aus diesem Grund sollten Sie die Verwendung von Verbindungspooling oder die Wiederverwendung vorhandener Redis OSS-Verbindungen in Betracht ziehen.

- **Skalierungsgeschwindigkeit:** ElastiCache Serverless skaliert automatisch, wenn Ihre Anforderungsrate steigt. Ein plötzlicher starker Anstieg der Anforderungsrate, der schneller ist als die Geschwindigkeit, mit der ElastiCache Serverless skaliert, kann für einige Zeit zu einer erhöhten Latenz führen. ElastiCache Serverless kann die unterstützte Anforderungsrate in der Regel schnell erhöhen. Es dauert bis zu 10 bis 12 Minuten, bis sich die Anforderungsrate verdoppelt.
- **Inspizieren Sie Befehle mit langer Laufzeit:** Einige Redis OSS-Befehle, einschließlich Lua-Skripten oder Befehle für große Datenstrukturen, können lange laufen. ElastiCache veröffentlicht Metriken auf Befehlsebene, um diese Befehle zu identifizieren. Mit [ElastiCache Serverless](#) können Sie die `BasedECPUs` Metriken verwenden.
- **Gedrosselte Anfragen:** Wenn Anfragen in ElastiCache Serverless gedrosselt werden, kann es zu einem Anstieg der clientseitigen Latenz in Ihrer Anwendung kommen. [Wenn Anfragen in Serverless gedrosselt werden, sollten Sie einen ElastiCache Anstieg der Serverless-Metrik feststellen. `ThrottledRequests` ElastiCache](#) Im folgenden Abschnitt finden Sie Informationen zur Behebung gedrosselter Anfragen.
- **Gleichmäßige Verteilung von Schlüsseln und Anfragen:** In ElastiCache (Redis OSS) kann eine ungleichmäßige Verteilung von Schlüsseln oder Anfragen pro Steckplatz zu einem Hot-Slot führen, was zu einer erhöhten Latenz führen kann. ElastiCache Serverless unterstützt bis zu 30.000 ECPUs/Sekunde (90.000 ECPUS/Sekunde bei Verwendung von Read from Replica) auf einem einzigen Steckplatz in einem Workload, der einfache SET/GET-Befehle ausführt. Wir empfehlen, Ihre Schlüssel- und Anforderungsverteilung auf die Slots zu überprüfen und für eine gleichmäßige Verteilung zu sorgen, falls Ihre Anforderungsrate diese Grenze überschreitet.

Behebung von Drosselungsproblemen in Serverless ElastiCache

In serviceorientierten Architekturen und verteilten Systemen wird die Begrenzung der Geschwindigkeit, mit der API-Aufrufe durch verschiedene Servicekomponenten verarbeitet werden, als Drosselung bezeichnet. Dadurch werden Leistungsspitzen geglättet, Abweichungen im Komponentendurchsatz vermieden und bei unerwarteten Betriebsereignissen besser vorhersehbare Wiederherstellungen ermöglicht. ElastiCache Serverless wurde für diese Art von Architekturen entwickelt, und die meisten Redis OSS-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 Anfragen in Serverless gedrosselt werden, sollten Sie einen Anstieg der ElastiCache Serverless-Metrik feststellen. ThrottledRequests ElastiCache](#) Wenn Sie eine hohe Anzahl gedrosselter Anfragen feststellen, sollten Sie Folgendes beachten:

- **Skalierungsgeschwindigkeit:** ElastiCache Serverless wird automatisch skaliert, wenn Sie mehr Daten aufnehmen oder Ihre Anforderungsrate erhöhen. Wenn Ihre Anwendung schneller skaliert als die Geschwindigkeit, mit der Serverless skaliert, werden Ihre Anfragen möglicherweise gedrosselt, während ElastiCache Serverless an Ihre Arbeitslast angepasst wird. ElastiCache ElastiCache Serverless kann die Speichergröße in der Regel schnell erhöhen. Es dauert bis zu 10 bis 12 Minuten, bis sich die Speichergröße in Ihrem Cache verdoppelt hat.
- **Gleichmäßige Verteilung von Schlüsseln und Anfragen:** In ElastiCache (Redis OSS) kann eine ungleichmäßige Verteilung von Schlüsseln oder Anfragen pro Steckplatz zu einem Hot-Slot führen. Ein Hot-Slot kann zu einer Drosselung von Anfragen führen, wenn die Anforderungsrate für einen einzelnen Steckplatz 30.000 ECPUS/Sekunde übersteigt, und das bei einer Arbeitslast, die einfache SET/GET-Befehle ausführt.
- **Aus Replikat lesen:** Wenn Ihre Anwendung dies zulässt, sollten Sie in Erwägung ziehen, die Funktion „Aus Replikat lesen“ zu verwenden. Die meisten Redis OSS-Clients können so konfiguriert werden, dass sie Lesevorgänge skalieren „, sodass Lesevorgänge direkt an Replikatknoten weitergeleitet werden. Mit dieser Funktion können Sie den Lesetraffic skalieren. Darüber hinaus leitet ElastiCache Serverless automatisch Lesevorgänge von Replikatanfragen an Knoten weiter, die sich in derselben Availability Zone wie Ihre Anwendung befinden, was zu einer geringeren Latenz führt. Wenn Read from Replica aktiviert ist, können Sie für Workloads mit einfachen SET/GET-Befehlen bis zu 90.000 ECPUS/Sekunde auf einem einzigen Steckplatz 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 OSS. Standardmäßig gestatten Sicherheitsgruppen allen ausgehenden Datenverkehr. In diesem Fall ist nur die eingehende Regel in der Zielsicherheitsgruppe erforderlich.

Weitere Informationen finden Sie unter [Zugriffsmuster für den Zugriff auf einen ElastiCache Cluster in 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, um den Redis OSS-Verkehr zuzulassen, 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 OSS ist eine Multithread-Anwendung. Die Ausführung jedes Befehls erfolgt jedoch in einem einzigen (Haupt-) Thread. Aus diesem Grund ElastiCache bietet es die Metriken `CPUUtilization` und `EngineCPUUtilization`. `EngineCPUUtilization` stellt die CPU-Auslastung für den Redis OSS-Prozess und `CPUUtilization` die Nutzung für alle vCPUs bereit. Knoten mit mehr als einer vCPU haben normalerweise unterschiedliche Werte für `CPUUtilization` und `EngineCPUUtilization`, die zweite ist allgemein höher.

`EngineCPUUtilization` 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 `EngineCPUUtilization`-Muster. Nützliche Metriken sind:
 - `CacheHits` und `CacheMisses`: Die Anzahl der erfolgreichen Anforderungen oder Anforderungen, die kein gültiges Element im Cache gefunden haben. Wenn das Verhältnis von Fehlern im Vergleich zu Treffern hoch ist, verschwendet die Anwendung Zeit und Ressourcen mit unfruchtbaren Anfragen.
 - `SetTypeCmds` und `GetTypeCmds`: Diese Metriken, die mit `EngineCPUUtilization` korreliert sind, können helfen zu verstehen, ob die Last für Schreibanforderungen signifikant höher ist, gemessen durch `SetTypeCmds`, oder liest, gemessen durch `GetTypeCmds`. Wenn die Last überwiegend Lesevorgänge ist, kann die Verwendung mehrerer Read-Replicas die Anforderungen über mehrere Knoten hinweg ausgleichen und die primäre für Schreibvorgänge ersparen. In Clustern mit deaktiviertem Clustermodus 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 und die Replikatknoten erkennen, die Lesevorgänge über den Befehl [READONLY Redis OSS aktivieren und die Leseanfragen](#) an die Replikate senden.
- Erhöhte Anzahl von Verbindungen:
 - `CurrConnections` und `NewConnections`: `CurrConnection` 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 `NewConnection` sollte so niedrig wie möglich sein. Redis OSS 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 OSS) mit aktiviertem Clustermodus, sodass die Schreibvorgänge auf mehrere Shards verteilt werden können, oder die Skalierung auf einen Knotentyp mit mehr Netzwerkkapazitäten.
- Die CloudWatchmetrics `NetworkBytesIn` und `NetworkBytesOut` geben die Datenmenge an, die in den Knoten eingeht bzw. den Knoten verlässt. `ReplicationBytes` ist der Verkehr, der der Datenreplikation gewidmet ist.

Weitere Informationen finden Sie unter [Netzwerkbezogene Grenzen](#).

- **Komplexe Befehle:** Redis OSS-Befehle werden in einem einzigen Thread bedient, was bedeutet, dass Anfragen sequentiell bedient werden. Ein einzelner langsamer Befehl kann sich auf andere Anforderungen und Verbindungen auswirken, was zu Timeouts führt. Die Verwendung von Befehlen, die auf mehrere Werte, Schlüssel oder Datentypen wirken, muss sorgfältig

durchgeführt werden. Verbindungen können abhängig von der Anzahl der Parameter oder der Größe der Ein- oder Ausgabewerte blockiert oder beendet werden.

Ein berüchtigtes Beispiel ist die KEYS-Befehl. Es fegt den gesamten Schlüsselraum auf der Suche nach einem bestimmten Muster und blockiert die Ausführung anderer Befehle während der Ausführung. Redis OSS verwendet die Notation „Big O“, um die Komplexität seiner Befehle zu beschreiben.

Keys Befehl hat $O(N)$ Zeitkomplexität, wobei N die Anzahl der Schlüssel in der Datenbank ist. Je größer die Anzahl der Schlüssel ist, desto langsamer wird der Befehl. KEYS kann auf verschiedene Arten Probleme verursachen: Wenn kein Suchmuster verwendet wird, gibt der Befehl alle verfügbaren Schlüsselnamen zurück. In Datenbanken mit tausend oder Millionen von Elementen wird eine riesige Ausgabe erstellt und die Netzwerkpuffer überflutet.

Wenn ein Suchmuster verwendet wird, werden nur die Schlüssel, die dem Muster entsprechen, an den Client zurückgegeben. Die Engine wird jedoch immer noch den gesamten Schlüsselraum durchsucht, und die Zeit, um den Befehl abzuschließen, ist gleich.

Eine Alternative für KEYS ist die SCAN-Befehl. Es iteriert über den Schlüsselraum und begrenzt die Iterationen in einer bestimmten Anzahl von Elementen, wodurch längere Blöcke auf der Engine vermieden werden.

Der Scan hat die COUNT, der zum Festlegen der Größe der Iterationsblöcke verwendet wird. Der Standardwert ist 10 (10 Elemente pro Iteration).

Abhängig von der Anzahl der Elemente in der Datenbank sind kleine COUNT-Werte-Blöcke erfordern mehr Iterationen, um einen vollständigen Scan abzuschließen, und größere Werte halten die Engine bei jeder Iteration länger beschäftigt. Während kleine Zählwerte SCAN langsamer in großen Datenbanken machen, können größere Werte die gleichen Probleme verursachen, die für KEYS beschrieben sind.

Als Beispiel wird das Ausführen des SCAN Befehl mit Zählwert als 10 erfordert 100.000 Wiederholungen in einer Datenbank mit 1 Million Schlüsseln. Wenn die durchschnittliche Netzwerk-Roundtrip-Zeit 0,5 Millisekunden beträgt, werden etwa 50.000 Millisekunden (50 Sekunden) für die Übertragung von Anforderungen ausgegeben.

Auf der anderen Seite, wenn der Zählwert 100,000 wäre, wäre eine einzelne Iteration erforderlich und nur 0,5 ms würden ausgegeben, um sie zu übertragen. Die Engine wäre jedoch

für andere Operationen vollständig blockiert, bis der Befehl den gesamten Schlüsselraum beendet hat.

Außerdem KEYS, sind einige andere Befehle potenziell schädlich, wenn sie nicht korrekt verwendet werden. Eine Liste aller Befehle und ihrer jeweiligen Zeitkomplexität finden Sie unter <https://redis.io/commands>.

Beispiele für mögliche Probleme:

- **Lua-Skripte:** Redis OSS bietet einen eingebetteten Lua-Interpreter, der die Ausführung von Skripten auf der Serverseite ermöglicht. Lua-Skripte auf Redis OSS werden auf Engine-Ebene ausgeführt und sind per Definition atomar, was bedeutet, dass kein anderer Befehl oder Skript ausgeführt werden darf, während ein Skript ausgeführt wird. Lua-Skripte bieten die Möglichkeit, mehrere Befehle, Entscheidungsalgorithmen, Datenanalyse und andere direkt auf der Redis OSS-Engine auszuführen. Während die Atomizität von Skripten und die Möglichkeit, die Anwendung zu entladen, verlockend sind, müssen Skripte mit Sorgfalt und für kleine Operationen verwendet werden. Bei ElastiCache aktivierter Option ist die Ausführungszeit von Lua-Skripten auf 5 Sekunden begrenzt. Skripte, die nicht in den Schlüsselraum geschrieben wurden, werden nach Ablauf der 5 Sekunden automatisch beendet. Um Datenbeschädigungen und Inkonsistenzen zu vermeiden, wird der Knoten ein Failover ausgeführt, wenn die Skriptausführung nicht innerhalb von 5 Sekunden abgeschlossen wurde und während der Ausführung Schreibvorgänge stattfand. [Transaktionen](#) sind die Alternative, um die Konsistenz mehrerer verwandter wichtiger Änderungen in Redis OSS 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, einem asynchronen Befehl, der seit Redis OSS 4 verfügbar ist. UNLINK muss, DEL wann immer möglich, vorgezogen werden. Ab ElastiCache (Redis OSS) 6.x sorgt der `lazyfree-lazy-user-del` Parameter dafür, dass sich der DEL Befehl so verhält, UNLINK als ob er aktiviert wäre. Weitere Informationen finden Sie unter [Redis OSS 6.0-Parameteränderungen](#).

- Befehle, die auf mehrere Tasten wirken: DEL wurde zuvor als Befehl erwähnt, der mehrere Argumente akzeptiert und seine Ausführungszeit wird direkt proportional dazu sein. Redis OSS bietet jedoch viele weitere Befehle, die ähnlich funktionieren. Als Beispiele MSET und MGET ermöglichen das gleichzeitige Einfügen oder Abrufen mehrerer String-Schlüssel. Ihre Nutzung kann vorteilhaft sein, um die Netzwerklatenz zu reduzieren, die mehreren einzelnen SET oder GET-Befehle. Eine umfangreiche Liste von Parametern wirkt sich jedoch auf die CPU-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 einwirken: Redis OSS bietet auch Befehle, die auf einen oder mehrere Schlüssel wirken, unabhängig von ihrem Datentyp. ElastiCache (Redis OSS) 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> und wählen Sie die folgenden Optionen aus:
 - Quelltyp: Wählen Sie die Instance, 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 AWS Lambda usw.) 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 ElastiCache (Redis OSS) oder 11211 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 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 `CurConnections` Metriken überwacht werden. CloudWatch Clients haben jedoch auch ihre Grenzen für ausgehende Verbindungen. Überprüfen Sie unter Linux den zulässigen flüchtigen Portbereich mit folgendem Befehl:

```
# sysctl net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 32768 60999
```

Im vorherigen Beispiel sind 28231 Verbindungen von derselben Quelle zu derselben Ziel-IP (ElastiCache Knoten) und demselben Zielport zulässig. Der folgende Befehl zeigt, wie viele Verbindungen für einen bestimmten ElastiCache Knoten (IP 1.2.3.4) bestehen:

```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
'^State' | wc -l
```

Wenn die Zahl zu hoch ist, wird Ihr System möglicherweise überlastet und versucht, die Verbindungsanforderungen zu verarbeiten. Es ist ratsam, Techniken wie Verbindungspooling oder persistente Verbindungen zu implementieren, um die Verbindungen besser zu handhaben. Wenn möglich, konfigurieren Sie den Verbindungspool so, dass die maximale Anzahl von Verbindungen auf einige hundert begrenzt wird. Außerdem wäre eine Back-Off-Logik zur Behandlung von Timeouts oder anderen Verbindungsausnahmen ratsam, um im Falle von Problemen eine Verbindungsabwanderung zu vermeiden.

- **Grenzwerte für den Netzwerkverkehr:** Überprüfen Sie die folgenden [CloudWatch Metriken für Redis OSS](#), um mögliche Netzwerklimits zu identifizieren, die auf dem Knoten erreicht werden: ElastiCache
 - `NetworkBandwidthInAllowanceExceeded/NetworkBandwidthOutAllowanceExceeded`: Netzwerkpakete geformt, weil der Durchsatz das aggregierte Bandbreitenlimit überschritten hat.

Es ist wichtig zu beachten, dass jedes Byte, das auf den primären Knoten geschrieben wird, auf N Replikate repliziert wird, wobei N die Anzahl der Replikate ist. Cluster mit kleinen Knotentypen, mehreren Replikaten und intensiven Schreibanforderungen 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 OSS SlowLogs**: In der ElastiCache Standardkonfiguration werden die letzten 128 Befehle beibehalten, deren Ausführung mehr als 10 Millisekunden gedauert hat. Der Verlauf der langsamen Befehle wird während der Motorlaufzeit beibehalten und geht im Falle eines Fehlers oder eines Neustarts verloren. Wenn die Liste 128 Einträge erreicht, werden alte Ereignisse entfernt, um Raum für neue zu öffnen. Die Größe der Liste der langsamen Ereignisse und die Ausführungszeit, die als langsam betrachtet wird, kann durch die Parameter `slowlog-max-len` und `slowlog-log-slower-than` in einer [Benutzerdefinierte Parametergruppe](#) angepasst werden. Die slowlogs-Liste kann abgerufen werden, indem `SLOWLOG GET 128` auf dem Motor, 128 sind die letzten 128 langsamen Befehle gemeldet. Jeder Spielereintrag hat folgende Felder:


```
1) 1) (integer) 1 -----> Sequential ID
   2) (integer) 1609010767 --> Timestamp (Unix epoch time)of the Event
   3) (integer) 4823378 -----> Time in microseconds to complete the command.
   4) 1) "keys" -----> Command
      2) "*" -----> Arguments
   5) "1.2.3.4:57004"-> Source
```

Das 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 (Redis OSS) bietet CloudWatch Metriken zur Überwachung der durchschnittlichen Latenz für verschiedene Befehlsklassen. 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, geoadd;
 - 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, sunion, 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 OSS-Laufzeitbefehle:**
 - **info commandstats:** Stellt eine Liste der Befehle bereit, die seit dem Start der Redis OSS-Engine ausgeführt wurden, sowie deren Gesamtzahl der Ausführungen, die Gesamtausführungszeit und die durchschnittliche Ausführungszeit pro Befehl;
 - **Client-Liste:** Bietet eine Liste der aktuell verbundenen Clients und relevante Informationen wie Pufferverwendung, zuletzt ausgeführter Befehl usw.;
- **Backup und Replikation:** ElastiCache (Redis OSS) -Versionen vor 2.8.22 verwenden einen Fork-Prozess, um Backups zu erstellen und vollständige Synchronisationen mit den Replikaten durchzuführen. Diese Methode kann in erheblichem Speicheraufwand für schreibintensive Anwendungsfälle auftreten.

Beginnend mit ElastiCache Redis OSS 2.8.22 wurde eine Sicherungs- und Replikationsmethode ohne Forkless 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 Standardkonfiguration ElastiCache (Redis OSS) sorgt dafür, dass die Client-Verbindungen auf unbestimmte Zeit hergestellt werden. In einigen Fällen kann eine Verbindungsbeendigung jedoch wünschenswert sein. Zum Beispiel:

- Fehler in der Client-Anwendung können dazu führen, dass Verbindungen vergessen und im Leerlauf gehalten werden. Dies wird als „Verbindungsleck“ bezeichnet und die Folge ist eine stetige Zunahme der Anzahl etablierter Verbindungen, die auf der `CurrentConnections`-Metrik beobachtet werden. Dieses Verhalten kann zu einer Überlastung des Clients oder ElastiCache der Seite führen. Wenn eine sofortige Korrektur von der Client-Seite aus nicht möglich ist, legen einige Administratoren in ihrer ElastiCache Parametergruppe einen Wert für das „Timeout“ fest. Das Timeout ist die Zeit in Sekunden, die erlaubt ist, dass Verbindungen im Leerlauf bestehen bleiben. Wenn der Client innerhalb dieses Zeitraums keine Anfrage einreicht, beendet die Redis OSS-Engine die Verbindung, sobald die Verbindung den Timeout-Wert erreicht. Kleine Zeitüberschreitungswerte können zu unnötigen Trennungen führen, und Clients müssen sie ordnungsgemäß behandeln und eine erneute Verbindung herstellen, was zu Verzögerungen führt.
- Der Speicher, der zum Speichern von Schlüsseln verwendet wird, wird für Clientpuffer freigegeben. Langsame Clients mit großen Anfragen oder Antworten benötigen möglicherweise eine beträchtliche Menge an Speicher, um ihre Puffer zu verarbeiten. Die Standardkonfigurationen ElastiCache (Redis OSS) schränken die Größe der regulären Client-Ausgabepuffer nicht ein. Wenn das `SymbolMemoryLimit` erreicht wird, versucht die Engine, Elemente zu vertreiben, um die Pufferverwendung zu erfüllen. Bei extrem wenig Arbeitsspeicher entscheidet sich ElastiCache (Redis OSS) möglicherweise dafür, die Verbindung von Clients zu trennen, die große Client-Ausgabepuffer verbrauchen, um Speicherplatz freizugeben und den Zustand des Clusters zu erhalten.

Es ist möglich, die Größe von Clientpuffern mit benutzerdefinierten Konfigurationen zu begrenzen, und Clients, die das Limit erreichen, werden getrennt. Clients sollten jedoch in der Lage sein, unerwartete Trennungen zu verarbeiten. Die Parameter für die Puffergröße für reguläre Clients sind die folgenden:

- `client-query-buffer-limit`: Maximale Größe einer einzelnen Eingabeanforderung;
- `client-output-buffer-limit-normal-soft-limit`: Soft-Limit für Client-Verbindungen. Die Verbindung wird beendet, wenn sie länger als die für `client-output-buffer-limit` - definierte Zeit in Sekunden über dem Soft-Limit bleibt `normal-soft-seconds` oder wenn sie das Hard-Limit erreicht;
- `client-output-buffer-limit-normal-soft-seconds`: Zulässige Zeit für Verbindungen, die den Wert von `client-output-buffer-limit` - überschreiten `normal-soft-limit`;
- `client-output-buffer-limit-normal-hard-limit`: Eine Verbindung, die dieses Limit erreicht, wird sofort beendet.

Neben den regulären Clientpuffern steuern die folgenden Optionen den Puffer für Replikatknoten und Pub/Sub (Publish/Sub) Clients:

- client-output-buffer-limit-replica-hard-limit;
- client-output-buffer-limit-replica-soft-seconds;
- client-output-buffer-limit-replica-hard-limit;
- client-output-buffer-limit-pubsub-soft-limit;
- client-output-buffer-limit-pubsub-soft-seconds;
- client-output-buffer-limit-pubsub-hard-limit;

Clientseitige Fehlerbehebung für Amazon EC2 Instances

Die Auslastung und Reaktionsfähigkeit auf der Client-Seite können sich auch auf die Anfragen an 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,sack0K,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,sack0K,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. Wenn wir denselben Redis PING OSS-Befehl verwenden, erhalten `strace` wir:

```

$ echo ping | strace -f -tttt -r -e trace=execve,socket,open,recvfrom,sendto
nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com (http://
example.xxxxxx.ng.0001.use1.cache.amazonaws.com/)

```


Sicherheit in Amazon ElastiCache

Die Sicherheit in der Cloud hat für AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sicherheit eine kritische Bedeutung hat.

Sicherheit gilt zwischen AWS und Ihnen eine geteilte Verantwortung. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS-Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für Amazon ElastiCache gelten, finden Sie unter [AWS-Services im Geltungsbereich nach Compliance-Programm](#).
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Dienst bestimmt, den Sie verwenden. 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 geteilten Verantwortung bei der Verwendung von Amazon ElastiCache anwenden. Die folgenden Themen veranschaulichen, wie Sie Amazon ElastiCache zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren auch, wie Sie andere AWS-Dienste verwenden, die Ihnen bei der Überwachung und Sicherung Ihrer Amazon ElastiCache-Ressourcen helfen.

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](#)

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 bei Amazon ElastiCache](#)

Datensicherheit bei Amazon ElastiCache

Um Ihre Daten zu schützen, bieten Amazon ElastiCache und Amazon EC2 Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten auf dem Server.

Amazon ElastiCache (Memcached) bietet auch Verschlüsselungsfunktionen für Daten in Caches, auf denen Memcached-Versionen 1.6.12 oder höher ausgeführt werden.

- 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.

Themen

- [ElastiCache Verschlüsselung bei der Übertragung \(\) TLS](#)
- [Verschlüsselung im Ruhezustand in ElastiCache](#)

ElastiCache Verschlüsselung bei der Übertragung () TLS

Um die Sicherheit Ihrer Daten zu gewährleisten, EC2 bieten Amazon ElastiCache und Amazon Mechanismen zum Schutz vor unbefugtem Zugriff auf Ihre Daten auf dem Server. Durch die Bereitstellung von Verschlüsselungsfunktionen während der Übertragung ElastiCache erhalten Sie ein Tool, mit dem Sie Ihre Daten schützen können, wenn sie von einem Ort zum anderen übertragen werden.

Für alle Serverless-Caches ist die Verschlüsselung während der Übertragung aktiviert. Bei selbst entworfenen Clustern können Sie die Verschlüsselung während der Übertragung auf einem Cache-Cluster aktivieren, indem Sie den Parameter `TransitEncryptionEnabled` auf `true` (CLI: `--transit-encryption-enabled`) setzen, wenn Sie den Cache-Cluster mithilfe der Operation `CreateCacheCluster` (CLI: `create-cache-cluster`) erstellen.

Themen

- [Übersicht über die Verschlüsselung während der Übertragung](#)
- [Bedingungen für die Verschlüsselung während der Übertragung](#)
- [Bewährte Methoden für die Verschlüsselung während der Übertragung](#)

- [Aktivieren von Verschlüsselung während der Übertragung](#)
- [Verbinden von Knoten mit aktivierter Verschlüsselung während der Übertragung unter Verwendung von Openssl](#)
- [Einen TLS Memcached-Client mit Java erstellen](#)
- [Einen TLS Memcached-Client erstellen mit PHP](#)

Übersicht über die Verschlüsselung während der Übertragung

Die ElastiCache Amazon-Verschlüsselung bei der Übertragung ist eine Funktion, mit der Sie die Sicherheit Ihrer Daten an den anfälligsten Stellen erhöhen können — wenn sie von einem Ort zum anderen übertragen werden. Da für die Verschlüsselung und Entschlüsselung der Daten an den Endpunkten eine gewisse Verarbeitung erforderlich ist, kann die Aktivierung der Verschlüsselung während der Übertragung einige Performance-Einbußen zufolge haben. Sie sollten Benchmark-Tests Ihrer Daten mit und ohne Verschlüsselung während der Übertragung durchführen, um die Performance-Einbußen bei Ihren Anwendungsfällen zu bestimmen.

ElastiCache Die Verschlüsselung bei der Übertragung implementiert die folgenden Funktionen:

- Verschlüsselte Clientverbindungen — Client-Verbindungen zu Cache-Knoten sind TLS verschlüsselt.
- Verschlüsselte Serververbindungen – Daten, die zwischen Knoten in einem Cluster übertragen werden, sind verschlüsselt.
- Serverauthentifizierung – Clients können die Verbindung zum richtigen Server authentifizieren.

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 auf Clustern mit der Memcached-Version 1.6.12 und höher unterstützt.
- Die Verschlüsselung bei der Übertragung unterstützt die Versionen 1.2 und 1.3 von Transport Layer Security (TLS).
- Die Verschlüsselung während der Übertragung wird nur für Cluster unterstützt, die in einem Amazon VPC ausgeführt werden.

- Die Verschlüsselung während der Übertragung wird für Replikationsgruppen, auf denen die folgenden Knotentypen ausgeführt werden, nicht unterstützt: M1, M2, M3, R3, T2.

Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

- Verschlüsselung während der Übertragung wird durch die Festlegung des `TransitEncryptionEnabled`-Parameters auf `true` ausdrücklich aktiviert.
- Sie können die Verschlüsselung während der Übertragung nur für einen Cluster aktivieren, wenn Sie den Cluster erstellen. Sie können die Verschlüsselung während der Übertragung nicht aktivieren und deaktivieren, indem Sie einen Cluster ändern.
- Stellen Sie sicher, dass Ihr Caching-Client TLS Konnektivität unterstützt und dass Sie sie in der Client-Konfiguration aktiviert haben.

Bewährte Methoden für die Verschlüsselung während der Übertragung

- Aufgrund der für die Verschlüsselung und Entschlüsselung der Daten an den Endpunkten erforderliche Verarbeitung, kann die Implementierung der Verschlüsselung während der Übertragung eine Verringerung der Performance zur Folge haben. Führen Sie einen Benchmark-Test der Verschlüsselung während der Übertragung im Vergleich zu keiner Verschlüsselung Ihrer eigenen Daten durch, um deren Auswirkungen auf die Performance Ihrer Implementierung zu ermitteln.
- Da das Erstellen neuer Verbindungen teuer sein kann, können Sie die Auswirkungen der Verschlüsselung bei der Übertragung auf die Leistung verringern, indem Sie Ihre Verbindungen beibehalten. TLS

Aktivieren von Verschlüsselung während der Übertragung

Treffen Sie zum Aktivieren der Verschlüsselung während der Übertragung beim Erstellen eines Memcached-Clusters mit der AWS -Managementkonsole die folgende Auswahl:

- Wählen Sie Memcached als Ihre Engine.
- Wählen Sie die Engine-Version 1.6.12 und höher.
- Wählen Sie unter Encryption in transit (Verschlüsselung während der Übertragung) Enable (Aktivieren) aus.

Informationen zum step-by-step Vorgang finden Sie unter [Erstellen eines Memcached-Clusters](#) (Konsole).

Verbinden von Knoten mit aktivierter Verschlüsselung während der Übertragung unter Verwendung von Openssl

Um auf Daten von ElastiCache (Memcached-) Knoten zuzugreifen, für die Verschlüsselung während der Übertragung aktiviert ist, müssen Sie Clients verwenden, die mit Secure Socket Layer (SSL) arbeiten. Sie können Openssl `s_client` auch auf Amazon Linux und Amazon Linux 2 verwenden.

Verwenden Sie Openssl `s_client`, um sich mit einem Memcached-Cluster zu verbinden, der mit Verschlüsselung während der Übertragung auf Amazon Linux 2 oder Amazon Linux aktiviert ist, wie folgt:

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

Einen TLS Memcached-Client mit Java erstellen

Um einen Client im TLS Modus zu erstellen, gehen Sie wie folgt vor, um den Client mit dem entsprechenden `SSLContext` zu initialisieren:

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
public class TLSDemo {
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

        // Store a data item for an hour.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

```
}
```

Einen TLS Memcached-Client erstellen mit PHP

Um einen Client im TLS Modus zu erstellen, gehen Sie wie folgt vor, um den Client mit dem entsprechenden zu initialisieren: SSLContext

```
<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Discovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 * information
 * about Auto Discovery and persistent-id.
 */

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example */
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
mode */
$tls_client = new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastError_message(), "\n";
    exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSContextConfig in memcached-api.php for all configurations */
```

```
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
it to your client.
* Note: These TLS context configurations will be applied to all the servers connected
to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

/* store the data for 60 seconds in the cluster.
* The client will decide which cache host will store this item.
*/
if($tls_client->set('key', 'value', 60)) {
    print "Successfully stored key\n";
} else {
    echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
    print "Successfully retrieved key\n";
} else {
    echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
```

Weitere Informationen zur Verwendung des PHP Clients finden Sie unter [Installieren des ElastiCache-Cluster-Clients-für-PHP](#).

Verschlüsselung im Ruhezustand in ElastiCache

Um Ihre Daten zu schützen, bieten Amazon ElastiCache und Amazon S3 verschiedene Möglichkeiten, den Zugriff auf Daten in Ihrem Cache einzuschränken. Weitere Informationen finden Sie unter [Amazon VPCs und ElastiCache Sicherheit](#) und [Identity and Access Management für Amazon ElastiCache](#).

- Festplatte bei Synchronisierung und Austausch

ElastiCache bietet standardmäßige (vom Service verwaltete) Verschlüsselung im Ruhezustand sowie die Möglichkeit, Ihre eigenen symmetrischen, vom Kunden verwalteten AWS KMS-Schlüssel im [AWS Key Management Service](#) (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 (vom Service verwaltet) ist die einzige Option, die in den Regionen GovCloud (USA) verfügbar ist.

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 vom Kunden verwalteten Schlüsseln von AWS KMS](#)
- [Aktivieren der Verschlüsselung im Ruhezustand](#)
- [Weitere Informationen finden Sie unter:](#)

Bedingungen für die Verschlüsselung im Ruhezustand

Bei der Planung der Implementierung ElastiCache der Verschlüsselung im Ruhezustand sollten Sie die folgenden Einschränkungen bei der ElastiCache Verschlüsselung im Ruhezustand berücksichtigen:

- Die Verschlüsselung im Ruhezustand wird nur für Serverless-Caches unterstützt.
- Die Option, den vom Kunden verwalteten Schlüssel für die Verschlüsselung im Ruhezustand zu verwenden, ist in den Regionen AWS GovCloud (us-gov-east-1 und us-gov-west -1) nicht verfügbar.

Verwendung von vom Kunden verwalteten Schlüsseln von 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 in Ihrem AWS Konto erstellen, besitzen und verwalten. Weitere Informationen finden Sie unter [AWS -KMS-Schlüssel](#) im Entwicklerhandbuch zum AWS -Schlüsselverwaltungsdienst. Die Schlüssel müssen in AWS KMS erstellt werden, bevor sie mit ElastiCache verwendet werden können.

Informationen zum Erstellen von AWS KMS-Stammschlüsseln finden Sie unter [Creating Keys](#) im AWS Key Management Service Developer Guide.

ElastiCache ermöglicht Ihnen die Integration mit AWS KMS. Weitere Informationen finden Sie unter [Verwendung von Berechtigungen](#) im Entwicklerhandbuch zum AWS -Schlüsselverwaltungsdienst. Es sind keine Kundenaktionen erforderlich, um die ElastiCache Amazon-Integration mit AWS KMS zu aktivieren.

Der `kms:ViaService` Bedingungsschlüssel beschränkt die Verwendung eines AWS KMS-Schlüssels (KMS-Schlüssel) auf Anfragen von bestimmten AWS Diensten. Um `kms:ViaService` mit zu verwenden ElastiCache, schließen Sie beide ViaService Namen in den Wert des Bedingungsschlüssels ein: `elasticache.AWS_region.amazonaws.com` und `unddax.AWS_region.amazonaws.com`. Weitere Informationen finden Sie unter [kms: ViaService](#).

Sie können [AWS CloudTrail](#) damit die Anfragen verfolgen, an die Amazon in AWS Key Management Service Ihrem Namen ElastiCache sendet. Alle API-Aufrufe, die sich auf vom Kunden verwaltete Schlüssel AWS Key Management Service beziehen, haben entsprechende CloudTrail Protokolle. Sie können sich auch die Zuschüsse ansehen, die ElastiCache generiert werden, wenn Sie den [ListGrants](#) KMS-API-Aufruf aufrufen.

- Wenn Sie den Schlüssel löschen oder [deaktivieren](#) und für den Schlüssel, den Sie zur Verschlüsselung eines Caches verwendet haben, [Genehmigungen zurückziehen](#), kann der Cache nicht wiederhergestellt werden. Mit anderen Worten, es kann nach einem Hardwarefehler nicht geändert oder wiederhergestellt werden. AWS KMS löscht Root-Schlüssel erst nach einer Wartezeit von mindestens sieben Tagen. Nach dem Löschen des Schlüssels können Sie einen anderen vom Kunden verwalteten Schlüssel verwenden, um eine Sicherungskopie für Archivierungszwecke zu erstellen.
- Bei der automatischen Schlüsselrotation bleiben die Eigenschaften Ihrer AWS KMS-Root-Schlüssel erhalten, sodass die Rotation keine Auswirkungen auf Ihre Fähigkeit hat, auf Ihre Daten zuzugreifen. ElastiCache Verschlüsselte ElastiCache Amazon-Caches unterstützen keine manuelle Schlüsselrotation, bei der ein neuer Root-Schlüssel erstellt und alle Verweise auf den alten Schlüssel aktualisiert werden. Weitere Informationen finden Sie unter [Rotation von AWS KMS-Schlüsseln](#) im AWS Key Management Service Developer Guide.
- Für die Verschlüsselung eines ElastiCache Caches mithilfe eines KMS-Schlüssels ist ein Grant pro Cache erforderlich. Diese Genehmigung gilt für die Lebensdauer des Caches.
- Weitere Informationen zu AWS KMS-Zuschüssen und -Limits finden Sie unter [Limits](#) im AWS Key Management Service Developer Guide.

Aktivieren der Verschlüsselung im Ruhezustand

Für alle Serverless-Caches ist die Verschlüsselung im Ruhezustand aktiviert.

Sie können die Verschlüsselung im Ruhezustand aktivieren, wenn Sie einen ElastiCache Cache erstellen. Sie können dies mit der AWS Management Console AWS CLI, oder der ElastiCache API tun.

Beim Erstellen eines Caches können Sie eine der folgenden Optionen auswählen:

- Standard – Diese Option verwendet serviceverwaltete Verschlüsselung im Ruhezustand.
- Vom Kunden verwalteter Schlüssel — Mit dieser Option können Sie die Schlüssel-ID/den ARN von AWS KMS für die Verschlüsselung im Ruhezustand angeben.

Informationen zum Erstellen von AWS KMS-Root-Schlüsseln finden Sie unter [Create Keys](#) im AWS Key Management Service Developer Guide

Inhalt

- [Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS Management Console](#)

Aktivierung der Verschlüsselung im Ruhezustand mithilfe der AWS 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 KMS-Schlüssel, der AWS-Eigentümer ist, zum Verschlüsseln von Daten verwendet. Um Ihren eigenen AWS KMS Schlüssel auszuwählen, treffen Sie die folgenden Auswahlen:

- Erweitern Sie den Abschnitt Standardeinstellungen.
- Wählen Sie im Abschnitt Standardeinstellungen die Option Standardeinstellungen anpassen aus.
- Wählen Sie im Abschnitt Sicherheit die Option Sicherheitseinstellungen anpassen aus.
- Wählen Sie unter der Einstellung Verschlüsselungsschlüssels die Option Kundenverwalteter CMK aus.
- Wählen Sie unter der Einstellung AWS KMS -Schlüssel einen Schlüssel aus.

Weitere Informationen finden Sie unter:

- [Amazon VPCs und ElastiCache Sicherheit](#)
- [Identity and Access Management für Amazon ElastiCache](#)

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-Classik 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](#).

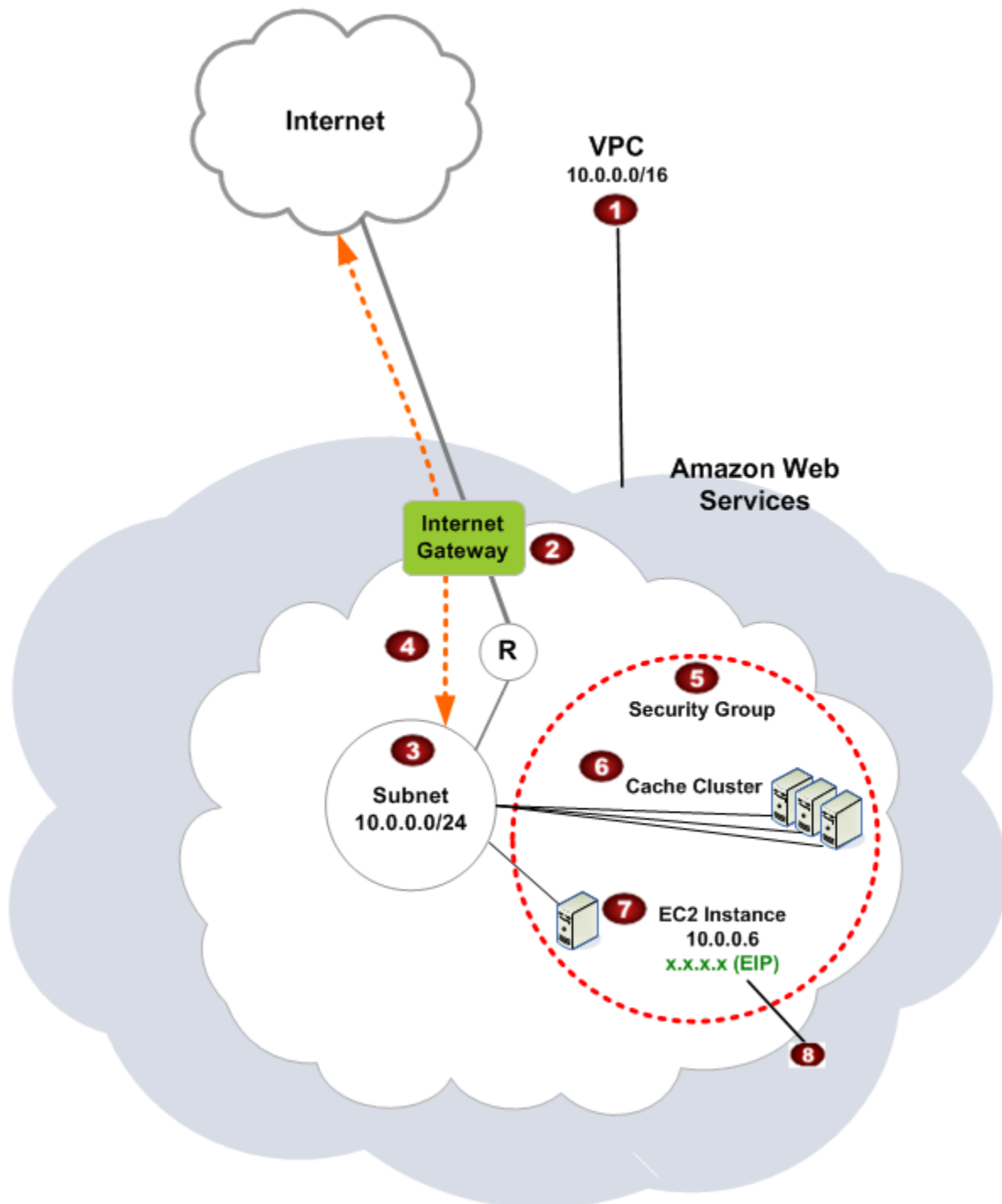
Mit AWS können Sie ein virtuelles Netzwerk in der Amazon Virtual Private Cloud erstellen, das einem herkömmlichen Rechenzentrum sehr ähnelt. Sie können Ihre Amazon VPC konfigurieren und den IP-Adressbereich auswählen, Subnetze erstellen sowie Routing-Tabellen, Netzwerk-Gateways und Sicherheitseinstellungen 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:

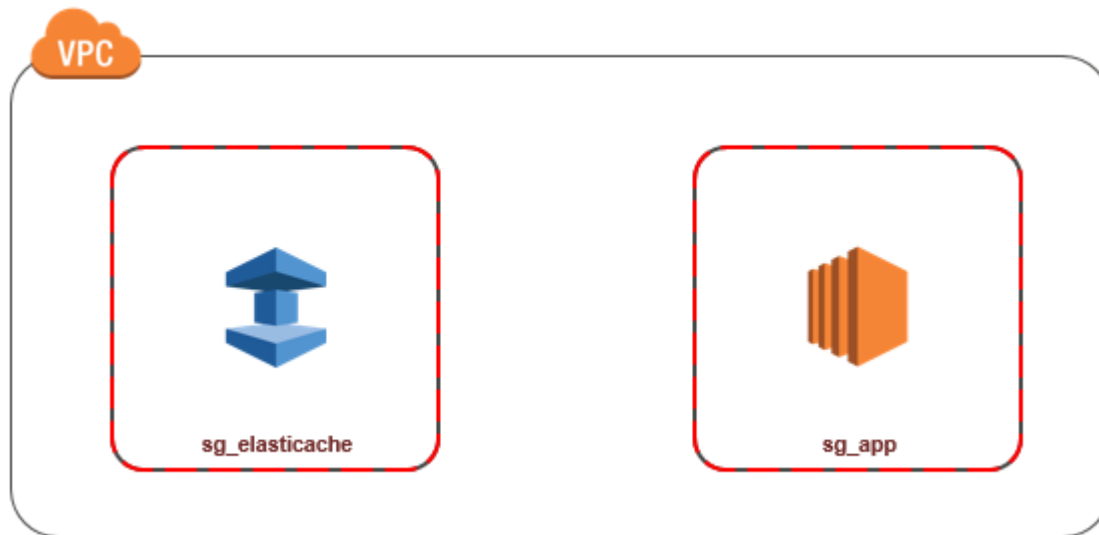
Inhalt

- [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 Standard-Port für Memcached-Caches ist 11211.

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 Memcached-Caches ist 11211.
 - 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.

The screenshot shows the 'Edit inbound rules' dialog in the AWS console. It has a title bar with 'Edit inbound rules' and a close button. Below the title bar are four columns: 'Type', 'Protocol', 'Port Range', and 'Source'. The 'Type' dropdown is set to 'Custom TCP Rule'. The 'Protocol' field contains 'TCP'. The 'Port Range' field contains '6379'. The 'Source' dropdown is set to 'Custom', and a search box next to it contains 'sg_app'. A dropdown menu is open below the search box, showing a list of security groups, with 'sg-99fc5ce6 - sg_app' selected. At the bottom left is an 'Add Rule' button. At the bottom right are 'Cancel' and 'Save' buttons.

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

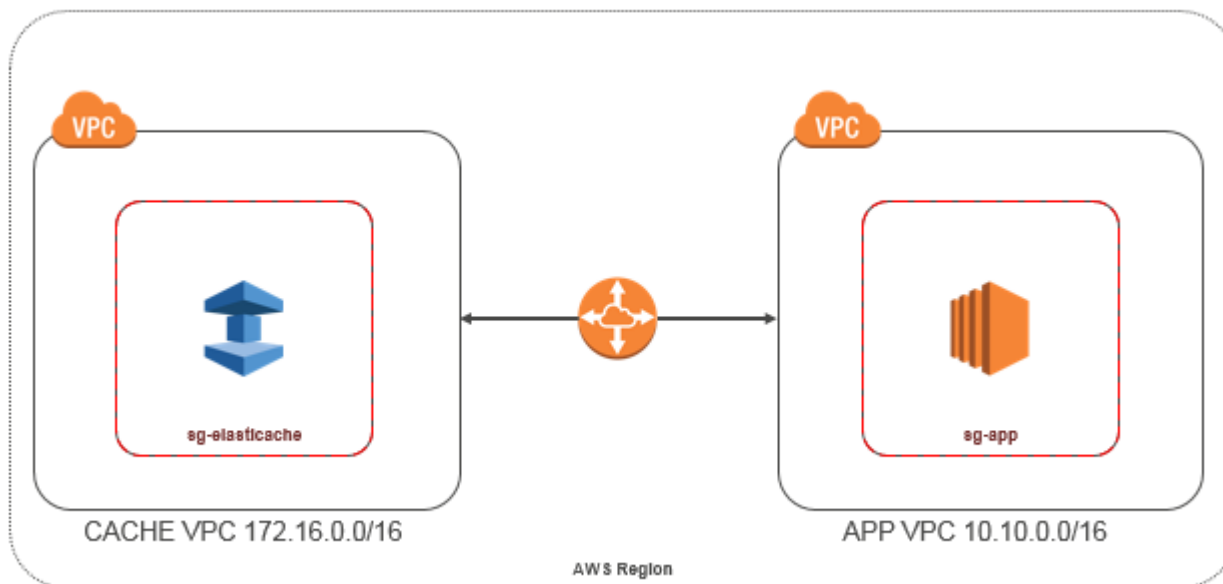
und die EC2-Instance in verschiedenen VPCs, aber in derselben Region befinden, können Sie VPC-Peering verwenden. Wenn sich der Cache und die EC2-Instance in verschiedenen Regionen befinden, können Sie VPN-Konnektivität zwischen den Regionen aufbauen.

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.

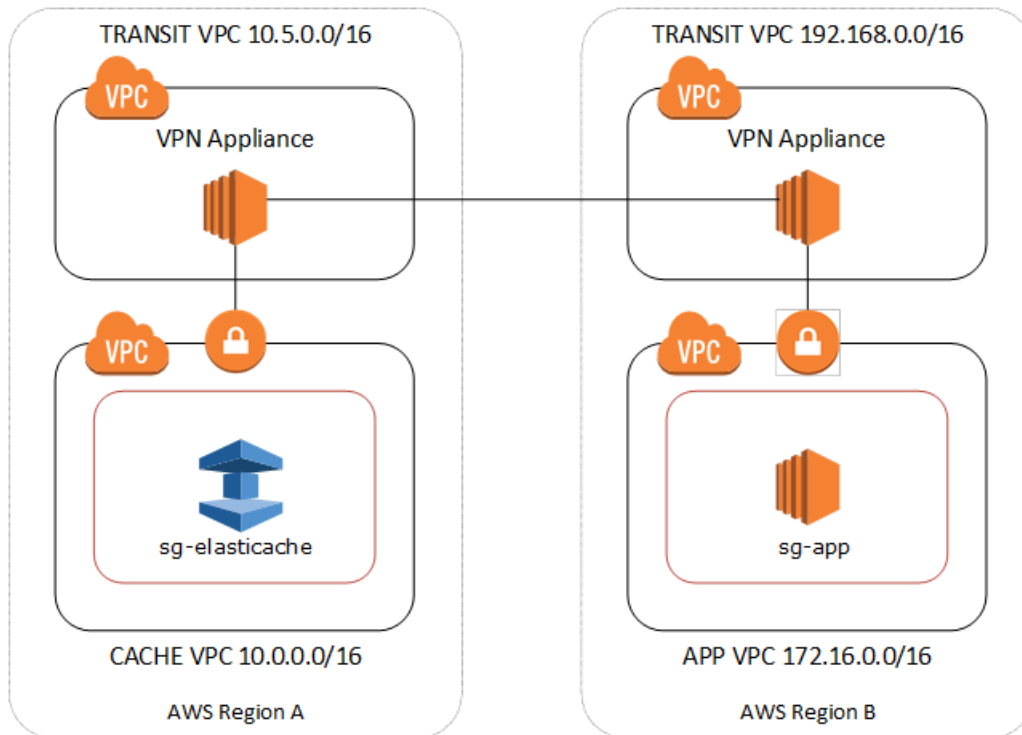
Sie können auch einen Peering-Verbindungsanhang zwischen Transit-Gateways in verschiedenen Regionen erstellen. AWS Auf diese Weise können Sie den Datenverkehr zwischen den Anhängen der Transit Gateways über verschiedene Regionen hinweg leiten.

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

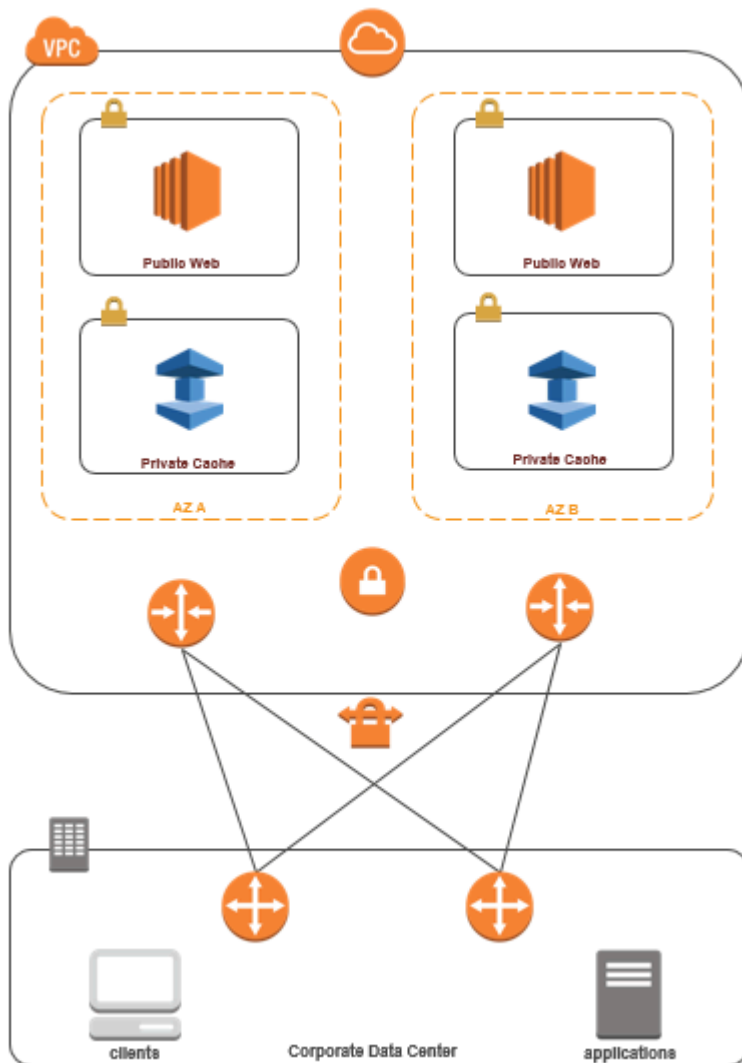
Ein anderes mögliches Szenario ist eine Hybridarchitektur, bei der Clients oder Anwendungen im Rechenzentrum des Kunden möglicherweise auf einen ElastiCache Cache in der VPC zugreifen müssen. Dieses Szenario wird auch unterstützt, vorausgesetzt, dass entweder über VPN oder Direct Connect Konnektivität 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

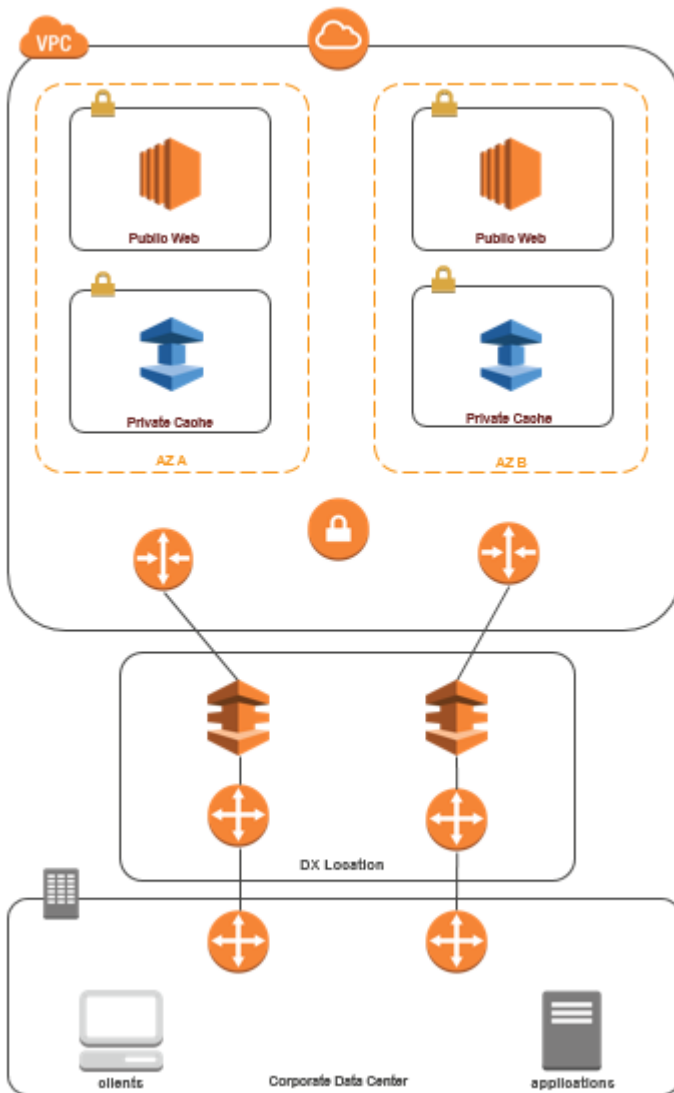
So greifen Sie auf einen Cache in einer VPC von einer Anwendung vor Ort über eine VPN-Verbindung zu

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.

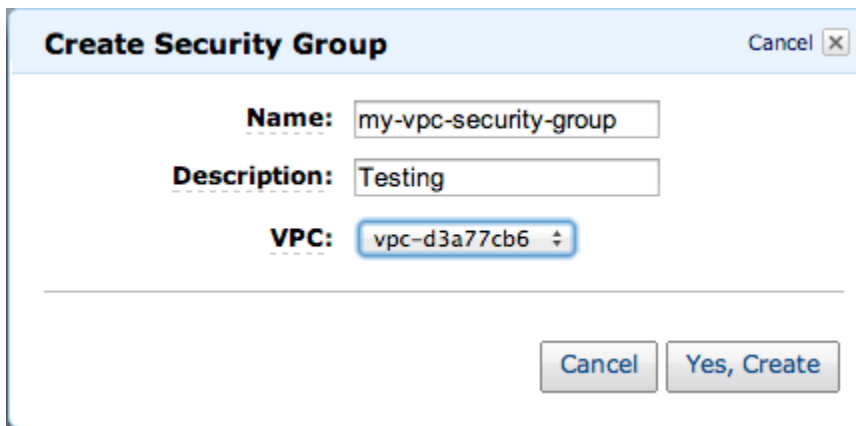
Erstellen einer Amazon VPC (Konsole)

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 Cancel

Name: my-vpc-security-group

Description: Testing

VPC: vpc-d3a77cb6

Cancel Yes, Create

- 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](#)
- [Erstellen eines Memcached-Clusters \(Konsole\)](#).

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 Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie 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-Architektur-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 sind jetzt bereit, mit Ihrem ElastiCache Cluster zu interagieren. Vorher aber müssen Sie, wenn Sie es noch nicht getan haben, das Dienstprogramm `telnet` installieren.

Installieren von `telnet` und Interaktion mit Ihrem Cache-Cluster (AWS CLI)

- Öffnen Sie ein Befehlsfenster. Führen Sie an der Eingabeaufforderung den folgenden Befehl aus. Geben Sie bei der Eingabeaufforderung zur Bestätigung `y` ein.

```
sudo yum install telnet
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB    00:00

...(output omitted)...

Complete!
```

2. Rufen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/> auf und rufen Sie den Endpunkt für einen der Knoten in Ihrem Cache-Cluster ab. Weitere Informationen finden Sie unter [Verbindungsendpunkte ermitteln](#) für Memcached.
3. Verwenden Sie telnet, um über Port 11211 eine Verbindung zu Ihrem Cache-Knotenendpunkt herzustellen. Ersetzen Sie den unten gezeigten Hostnamen durch den Hostnamen Ihres Cache-Knotens.

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

Sie sind nun mit der Cache-Engine verbunden und können Befehle ausgeben. In diesem Beispiel fügen Sie dem Cache ein Datenelement hinzu und rufen es dann sofort danach ab. Schließlich trennen Sie die Verbindung zum Cache-Knoten.

Um einen Schlüssel und einen Wert zu speichern, geben Sie die folgenden zwei Zeilen ein:

```
add mykey 0 3600 28  
This is the value for mykey
```

Die Cache-Engine antwortet wie folgt:

```
OK
```

So rufen Sie den Wert für mykey. Geben Sie für den Wert Folgendes ein:

```
get mykey
```

Die Cache-Engine antwortet wie folgt:

```
VALUE mykey 0 28  
This is the value for my key  
END
```

Geben Sie Folgendes ein, um die Verbindung zum Cache-Modul zu trennen:

```
quit
```

⚠ Important

Um zusätzliche Gebühren für Ihr AWS Konto zu vermeiden, sollten Sie alle AWS Ressourcen löschen, die Sie nicht mehr benötigen, nachdem Sie diese Beispiele ausprobiert haben.

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-Endpunktrichtlinien 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-Endpunktrichtlinie für die Amazon-ElastiCache-API

Sie können eine Endpunktrichtlinie 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"
    ],
    "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](#) .

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.

- Memcached-Cluster – Informationen zum Starten eines Memcached-Clusters finden Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#). Wählen Sie in Schritt 7.a (Advanced Memcached 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 AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um ElastiCache Ressourcen zu verwenden. IAMist eine AWS -Service , die Sie ohne zusätzliche Kosten verwenden können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So ElastiCache arbeitet Amazon mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)
- [Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff](#)
- [Zugriffskontrolle](#)
- [Übersicht über die Verwaltung von Zugriffsberechtigungen auf Ihre ElastiCache-Ressourcen](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie arbeiten ElastiCache.

Dienstbenutzer — Wenn Sie den ElastiCache Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr ElastiCache Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie nicht auf eine Funktion zugreifen können ElastiCache, finden Sie weitere Informationen unter [Fehlerbehebung Amazon ElastiCache Amazon-Identität und -Zugriff](#).

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die ElastiCache Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf ElastiCache. Es ist Ihre Aufgabe, zu bestimmen, auf welche ElastiCache Funktionen und Ressourcen Ihre Servicebenutzer

zugreifen sollen. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die Berechtigungen Ihrer Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehen IAM. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit verwenden kann ElastiCache, finden Sie unter [So ElastiCache arbeitet Amazon mit IAM](#).

IAM Administrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff darauf zu verwalten ElastiCache. Beispiele für ElastiCache identitätsbasierte Richtlinien, die Sie in verwenden können IAM, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM Benutzer authentifizieren (angemeldet bei AWS) oder indem Sie eine IAM Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center-) Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie AWS mithilfe eines Verbunds darauf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportale anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit der Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM Benutzerhandbuch unter AWS API Anfragen signieren](#).

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center

Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) AWS im IAM](#) Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein neues AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS -Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter [Aufgaben, für die Root-Benutzeranmeldedaten erforderlich](#) sind. IAM

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS -Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS -Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAMBenutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen

Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich](#) sind.

Eine [IAMGruppe](#) ist eine Identität, die eine Sammlung von IAM Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdmins und dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Wann sollte ein IAM Benutzer \(statt einer Rolle\) erstellt werden?](#) im IAMBenutzerhandbuch.

IAMRollen

Eine [IAMRolle](#) ist eine Identität innerhalb von Ihrem AWS-Konto, für die bestimmte Berechtigungen gelten. Sie ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Sie können vorübergehend eine IAM Rolle in der übernehmen, AWS Management Console indem Sie die [Rollen wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI AWS API OR-Operation aufrufen oder eine benutzerdefinierte Operation verwenden URL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie [unter Verwenden von IAM Rollen](#) im IAMBenutzerhandbuch.

IAMRollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle für einen externen Identitätsanbieter](#). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.

- Temporäre IAM Benutzerberechtigungen — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- Kontoübergreifender Zugriff — Sie können eine IAM Rolle verwenden, um einer Person (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS -Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie [IAM Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#). IAM
- Serviceübergreifender Zugriff — Einige AWS -Services verwenden Funktionen in anderen. AWS -Services Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- Zugriffssitzungen weiterleiten (FAS) — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Berechtigungen des Prinzipals, der an aufruft AWS -Service, kombiniert mit der Anforderung, Anfragen AWS -Service an nachgelagerte Dienste zu stellen. FASANfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS -Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- Servicerolle — Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle zum Delegieren von Berechtigungen AWS -Service an eine](#).
- Dienstbezogene Rolle — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS -Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance

ausgeführt werden und AWS API Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt](#) werden.

Informationen darüber, ob Sie IAM Rollen oder IAM Benutzer verwenden sollten, finden [Sie im Benutzerhandbuch unter Wann sollte eine IAM Rolle \(anstelle eines IAM Benutzers\) erstellt](#) werden.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS Form von JSON Dokumenten gespeichert. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter [Überblick über JSON Richtlinien](#).

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAMRichtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen aus dem AWS Management Console AWS CLI, dem oder dem abrufen AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAM Richtlinien erstellen im Benutzerhandbuch](#). IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können. AWS-Konto Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie wählen können, finden Sie im IAM Benutzerhandbuch unter [Auswahl zwischen verwalteten Richtlinien und Inline-Richtlinien](#).

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS -Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien nicht IAM in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten (ACLs)

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3 und AWS WAF Amazon VPC sind Beispiele für Dienste, die Unterstützung bieten ACLs. Weitere Informationen finden Sie unter [Übersicht über ACLs die Zugriffskontrollliste \(ACL\)](#) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAMBenutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAMBenutzerhandbuch unter [Berechtigungsgrenzen für IAM Entitäten](#).
- **Dienststeuerungsrichtlinien (SCPs)** — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Geräte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter [Richtlinien zur Servicesteuerung](#) im AWS Organizations Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Sitzungsrichtlinien](#).

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt

wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAMBenutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

So ElastiCache arbeitet Amazon mit IAM

Informieren Sie sich vor der Verwendung IAM zur Verwaltung des Zugriffs auf ElastiCache, welche IAM Funktionen zur Nutzung verfügbar sind ElastiCache.

IAMFunktionen, die Sie mit Amazon verwenden können ElastiCache

IAMFunktion	ElastiCache Unterstützung
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 wie andere AWS Dienste mit den meisten IAM Funktionen funktionieren, finden Sie IAM im IAMBenutzerhandbuch unter [AWS Dienste, die mit funktionieren](#).

Identitätsbasierte Richtlinien für ElastiCache

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter [IAM Richtlinien erstellen im Benutzerhandbuch](#). IAM

Mit IAM identitätsbasierten Richtlinien können Sie zulässige oder verweigernde Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie im IAM Benutzerhandbuch unter [Referenz zu IAM JSON Richtlinienelementen](#).

Beispiele für identitätsbasierte Richtlinien für ElastiCache

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Ressourcenbasierte Richtlinien in ElastiCache

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS -Services

Um den kontoübergreifenden Zugriff zu ermöglichen, können Sie in einer ressourcenbasierten Richtlinie ein ganzes Konto oder IAM Entitäten in einem anderen Konto als Prinzipal angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource

unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalidentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource gewähren. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie [IAMim IAMBenutzerhandbuch unter Kontoübergreifender Ressourcenzugriff](#).

Politische Maßnahmen für ElastiCache

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur eine Genehmigung erforderlich ist und für die es keinen entsprechenden Vorgang gibt. API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der ElastiCache Aktionen finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#) in der Service Authorization Reference.

Bei den in der Richtlinie ElastiCache verwendeten Aktionen wird vor der Aktion das folgende Präfix verwendet:

```
elasticache
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [  
  "elasticache:action1",  
  "elasticache:action2"  
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "elasticache:Describe*"
```

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Politische Ressourcen für ElastiCache

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem [Amazon-Ressourcennamen \(ARN\)](#) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*" 
```

Eine Liste der ElastiCache Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Von Amazon definierte Ressourcen ElastiCache](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Richtlinien-Bedingungsschlüssel für ElastiCache

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM Benutzer nur dann Zugriff auf eine Ressource gewähren, wenn sie mit seinem IAM Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [IAMRichtlinienelemente: Variablen und Tags](#).

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontext-Schlüssel für AWS globale Bedingungen](#) im IAMBenutzerhandbuch.

Eine Liste der ElastiCache Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon definierte Aktionen ElastiCache](#).

Beispiele für ElastiCache identitätsbasierte Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache](#)

Zugriffskontrolllisten (ACLs) in ElastiCache

UnterstütztACLs: Ja

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle (ABAC) mit ElastiCache

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen definiert. In AWS werden diese Attribute als Tags bezeichnet. Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABAC ist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu finden Sie ABAC unter [Was ist? ABAC](#) im IAM Benutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter [Verwenden der attributbasierten Zugriffskontrolle \(ABAC\)](#). IAM

Temporäre Anmeldeinformationen verwenden mit ElastiCache

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS -Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen darüber, AWS -Services wie Sie mit temporären Anmeldeinformationen [arbeiten können AWS -Services](#), finden Sie IAM im IAM Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Kennwort anmelden. Wenn

Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAMBenutzerhandbuch.

Mit dem AWS CLI oder können Sie manuell temporäre Anmeldeinformationen erstellen AWS API. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen unter IAM](#).

Serviceübergreifende Prinzipal-Berechtigungen für ElastiCache

Unterstützt Forward-Access-Sitzungen (FAS): Ja

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Berechtigungen des Prinzipals, der einen aufruft AWS -Service, kombiniert mit der Anforderung, Anfragen AWS -Service an nachgelagerte Dienste zu stellen. FASANfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS -Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für ElastiCache

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine [IAMRolle](#), die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter [Erstellen einer Rolle zum Delegieren von Berechtigungen AWS -Service an eine](#).

Warning

Das Ändern der Berechtigungen für eine Servicerolle kann zu ElastiCache Funktionseinschränkungen führen. Bearbeiten Sie Servicerollen nur, ElastiCache wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für ElastiCache

Unterstützt dienstbezogene Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS - Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen finden Sie unter [AWS Dienste, die mit funktionieren](#). IAM Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für Amazon ElastiCache

Standardmäßig 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-AWSverwaltete` 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:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

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 ElastiCache Ressourcen erstellen oder darauf zugreifen. Sie benötigen beispielsweise die erforderlichen Berechtigungen, um einen ElastiCache Cluster zu erstellen.

In den folgenden Abschnitten wird beschrieben, wie Sie Berechtigungen für verwalten ElastiCache. Wir empfehlen Ihnen, zunächst die Übersicht zu lesen.

- [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-Network-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:DescribeCacheClusters",
      "elasticache:ModifyServerlessCache",
      "elasticache:ModifyCacheCluster"
    ]
  }
],
```



```
    "Resource": "*"
  },
  {
    "Sid": "AllowUserToPassRole",
    "Effect": "Allow",
    "Action": [ "iam:PassRole" ],
    "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
  }
]
```

Die Richtlinie enthält zwei Anweisungen:

- Die erste Anweisung 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-API `CreateServiceLinkedRole` 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-API `CreateServiceLinkedRole` aufzurufen](#)

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 Systemadministratöraufgaben. 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-API `CreateServiceLinkedRole` 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"
        }
      }
    }
  ]
}
```

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 können Sie das `Condition` Element einer JSON-Richtlinie verwenden ElastiCache, um Schlüssel im

Anforderungskontext mit Schlüsselwerten zu vergleichen, die Sie in Ihrer Richtlinie angeben. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#).

Eine Liste der ElastiCache Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für Amazon ElastiCache](#) in der Service Authorization Reference.

Eine Liste der globalen Bedingungsschlüssel finden Sie unter [AWS Globale Bedingungskontextschlüssel](#).

Festlegung von Bedingungen: Verwenden von Bedingungsschlüsseln

Um eine differenzierte Steuerung zu implementieren, schreiben Sie eine IAM-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"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
```

```

        "StringEquals": {
            "elasticache:CacheNodeType": [
                "cache.r5.large"
            ]
        }
    }
}
]
}

```

Weitere Informationen finden Sie unter [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 enthält Beispielrichtlinien für die Implementierung einer detaillierten Zugriffskontrolle für die zuvor aufgeführten Parameter. ElastiCache

1. `elasticache:MaximumDataStorage`: Geben Sie den maximalen Datenspeicher eines serverlosen Caches an. Unter den angegebenen Bedingungen kann der Kunde keine Caches erstellen, die mehr als eine bestimmte Datenmenge speichern können.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ]
    }
  ]
}

```

```

    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
    ],
    "Condition": {
        "NumericLessThanEquals": {
            "elasticache:MaximumDataStorage": "30"
        },
        "StringEquals": {
            "elasticache:DataStorageUnit": "GB"
        }
    }
}
]
}

```

2. `elasticache:MaximizeCPUTerSecond`: Geben Sie den maximalen Wert für ECPU pro Sekunde eines serverlosen Caches 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`: Gibt an, welche `NodeType` (n) ein Benutzer erstellen kann. Unter Verwendung der bereitgestellten Bedingungen kann der Kunde einen einzelnen Wert oder einen Bereichswert für einen Knotentyp angeben.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:CacheNodeType": [
            "cache.t2.micro",
            "cache.t2.medium"
          ]
        }
      }
    }
  ]
}

```

```

    }
  ]
}

```

4. elasticache:EngineVersion: Spezifiziert die Verwendung der Engine-Version 1.6.6

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:EngineVersion": "1.6.6"
        }
      }
    }
  ]
}

```

5. elasticache:KmsKeyId: Geben Sie die Verwendung von vom Kunden verwalteten KMS-Schlüsseln an. AWS

```

{
  "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"
    }
  }
}
]
}

```

6. `elasticache:CacheParameterGroupName`: Geben Sie eine nicht standardmäßige Parametergruppe mit spezifischen Parametern von einer Organisation in Ihren Clustern an. Sie können auch ein Benennungsmuster für Ihre Parametergruppen angeben oder das Löschen eines bestimmten Parametergruppennamens blockieren. Im Folgenden finden Sie ein Beispiel, das die Verwendung von nur `"my-org-param-group"` einschränkt.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ]
    }
  ]
}

```

```

    ],
    "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
},

{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:CacheParameterGroupName": "my-org-param-group"
        }
    }
}
]
}

```

7. elasticacheCreateCacheCluster: CreateCacheCluster Aktion wird verweigert, wenn das Anforderungs-Tag Project fehlt oder nicht gleich oder ist. Dev QA Prod

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {

```

```

    "Effect": "Deny",
    "Action": [
      "elasticache:CreateCacheCluster"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
      "Null": {
        "aws:RequestTag/Project": "true"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:AddTagsToResource"
    ],
    "Resource": "arn:aws:elasticache:*:*:cluster:*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/Project": [
          "Dev",
          "Prod",
          "QA"
        ]
      }
    }
  }
]
}

```

8. elasticache:: Zulassen mit cache.r5.large oder cache.r6g.4xlarge und Tag. CacheNodeType CreateCacheCluster cacheNodeType Project=XYZ

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ]
    }
  ]
}

```

```
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:parametergroup:*",
      "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
      "StringEqualsIfExists": {
        "elasticache:CacheNodeType": [
          "cache.r5.large",
          "cache.r6g.4xlarge"
        ]
      },
      "StringEquals": {
        "aws:RequestTag/Project": "XYZ"
      }
    }
  }
]
```

Note

Beim Erstellen von Richtlinien zum gemeinsamen Erzwingen von Tags und anderen Bedingungsschlüsseln kann die Bedingung `IfExists` aufgrund der zusätzlichen `elasticache:AddTagsToResource`-Richtlinienanforderungen für Erstellungsanforderungen mit dem `--tags`-Parameter für Bedingungsschlüsselemente erforderlich sein.

Verwenden von serviceverknüpften Rollen für Amazon ElastiCache

Amazon ElastiCache verwendet 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
```

- 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 serviceverknüpfte 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) zugeordnet sind.

So überprüfen Sie in der IAM-Konsole, ob die serviceverknüpfte Rolle über eine aktive Sitzung verfügt

- Melden Sie sich bei der AWS Management Console an, und öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
- 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 der AWS Management Console](#)
 - [Verwendung der AWS CLI](#)
 - [Verwenden der 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 Bestätigungsdialogfeld 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 Löschaufgabe zu überprüfen. Geben Sie Folgendes ein, um eine Anforderung zum Löschen einer serviceverknüpften Rolle abzusenden.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. Geben Sie Folgendes ein, um den Status der Löschaufgabe zu überprüfen.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Der Status der Löschaufgabe kann `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` oder `FAILED` lauten. Wenn die Löschung fehlschlägt, gibt der Aufruf den Grund zurück, sodass Sie das Problem beheben können.

Löschen einer serviceverknüpften Rolle (IAM-API)

Sie können die 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 Dienstleistungen im Rahmen mehrerer AWS Compliance-Programme, wie z. B. SOC PCIRAMP, Fed undHIPAA.

Um zu erfahren, ob AWS -Service ein [AWS -Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS -Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter 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 Anwendungen erstellen HIPAA können, die AWS für sie in Frage kommen.

 Note

Nicht alle sind berechtigt AWS -Services . HIPAA Weitere Informationen finden Sie in der [Referenz für HIPAA qualifizierte Dienste](#).

- [AWS Ressourcen zur AWS](#) von Vorschriften — Diese Sammlung von Arbeitsmapen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [AWS Leitfäden zur Einhaltung von Vorschriften für Kunden](#) — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS -Services und die Leitlinien für Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zusammengefasst.
- [Evaluierung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Dies AWS -Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [Amazon GuardDuty](#) — Dies AWS -Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen zu

erfüllen PCIDSS, z. B. durch die Erfüllung der Anforderungen zur Erkennung von Eindringlingen, die in bestimmten Compliance-Frameworks vorgeschrieben sind.

- [AWS Audit Manager](#)— Auf diese AWS -Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Weitere Informationen

Allgemeine Informationen zur AWS Cloud-Compliance finden Sie im Folgenden:

- [FIPSEndpunkte nach Dienst](#)
- [Service-Updates in ElastiCache](#)
- [AWS Cloud-Konformität](#)
- [Modell der übergreifenden Verantwortlichkeit](#)
- [AWS PCIDSSCompliance-Programm](#)

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 Memcached](#)
- [Empfehlungen](#)

Minimieren von Ausfällen mit Memcached

Wenn Sie die Memcached-Engine verwenden, haben Sie folgende Möglichkeiten, um die Auswirkungen von Ausfällen möglichst gering zu halten. Es gibt zwei Arten von Ausfällen, die es zu berücksichtigen gilt: Knotenausfälle und Ausfälle einzelner Availability Zones.

Minimieren von Knotenausfällen

Serverless-Caches minimieren Knotenausfälle automatisch mit einer replizierten Multi-AZ-Architektur, sodass Knotenausfälle für Ihre Anwendung transparent sind. Sie sollten Ihre Cache-Daten auf mehrere Knoten verteilen, um die Auswirkungen von Knotenausfällen möglichst gering zu halten. Da selbst entworfene Cluster die Replikation nicht unterstützen, führt ein Knotenausfall zwangsläufig zu Datenverlusten in Ihrem Cluster.

Wenn Sie Ihren Memcached-Cluster erstellen, können Sie ihn mit 1 bis 60 Knoten oder mehr auf Anfrage erstellen. Wenn Sie Ihre Daten über mehrere Knoten partitionieren, gehen bei einem Knotenausfall weniger Daten verloren. Angenommen, Sie partitionieren Ihre Daten über 10 Knoten und ein einzelner Knoten speichert daher etwa 10 % der Cache-Daten. Bei einem Knotenausfall gehen in diesem Fall etwa 10 % Ihres Caches verloren und muss ersetzt werden, wenn ein neuer Knoten erstellt und bereitgestellt wird. Wenn dieselben Daten auf drei größere Knoten verteilt werden, gehen dagegen bei einem Knotenausfall etwa 33 % Ihrer Cache-Daten verloren.

[Wenn Sie mehr als 60 Knoten in einem Memcache-Cluster oder insgesamt mehr als 300 Knoten in einer AWS Region benötigen, füllen Sie das Formular zur Anforderung einer ElastiCache Limiterhöhung unter <https://aws.amazon.com/contact-us//aus.elasticache-node-limit-request>](https://aws.amazon.com/contact-us//aus.elasticache-node-limit-request)

Weitere Informationen zum Festlegen der Knotenanzahl in einem Memcached-Cluster erhalten Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

Minimieren von Ausfällen einer Availability Zone

Serverless-Caches minimieren Ausfälle von Availability Zones automatisch mit einer replizierten Multi-AZ-Architektur, sodass AZ-Ausfälle für Ihre Anwendung transparent sind.

Verteilen Sie Ihre Knoten auf möglichst viele Availability Zones, um die Auswirkungen von Ausfällen einer Availability Zone in einem selbst entworfenen Cluster möglichst gering zu halten. Sollte tatsächlich eine AZ ausfallen, verlieren Sie die Cache-Daten dieser AZ, nicht aber die Cache-Daten der anderen AZs.

Warum so viele Knoten?

Wenn meine Region nur 3 Availability Zones bereitstellt, warum sollte ich dann mehr als 3 Knoten erstellen, da im Fall eines AZ-Ausfalls sowieso etwa ein Drittel meiner Daten verloren geht?

Das ist eine sehr gute Frage. Dabei gilt es zu bedenken, dass wir zwei unterschiedliche Ausfalltypen minimieren wollen, Ausfälle einzelner Knoten und Availability Zones. Wenn Daten auf mehrere Availability Zones verteilt werden und eine der Zonen ausfällt, gehen nur die Cache-Daten dieser AZ verloren, unabhängig von der Anzahl der Knoten. Wenn jedoch ein Knoten ausfällt, gehen weniger Daten verloren, je mehr Knoten Sie verwenden.

Es gibt keine "Zauberformel", anhand derer sich die beste Anzahl an Knoten pro Cluster bestimmen lässt. Wägen Sie die Auswirkungen von Datenverlusten gegen die Wahrscheinlichkeit eines Ausfalls und die Kosten ab und treffen Sie anhand dieser Faktoren eine Entscheidung.

Weitere Informationen zum Festlegen der Knotenanzahl in einem Memcached-Cluster erhalten Sie unter [Erstellen eines Memcached-Clusters \(Konsole\)](#).

Weitere Informationen zu Regionen und Availability Zones finden Sie unter [Regionen und Availability Zones](#).

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

Wenn Sie Memcached ausführen und die Daten über mehrere Knoten partitionieren, sinkt die Größe des Datenverlustes beim Ausfall eines Knotens mit steigender Anzahl verwendeter Knoten.

Minimieren der Auswirkungen von Ausfällen einer Availability Zone

Knoten sollten auf möglichst viele Availability Zones verteilt werden, um die Auswirkungen von Ausfällen einer Availability Zone gering zu halten. 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.

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 verfügbar oder geplant verfügen.

Weitere Informationen zur Überprüfung und Installation von dienstspezifischen Updates auf die entsprechenden ElastiCache Cluster finden Sie unter [Anwenden der Service-Updates mithilfe der Konsole](#)

Wenn ein neues Service-Update für einen oder mehrere Ihrer ElastiCache Cluster verfügbar ist, können Sie die ElastiCache Konsole oder verwendenAPI, AWS CLI um das Update anzuwenden. Die folgenden Abschnitte beschreiben die Optionen, die zum Anwenden von Updates genutzt werden können.

Anwenden der Service-Updates mithilfe der Konsole

Um die Liste der Service-Updates und weitere Informationen anzuzeigen, gehen Sie zur Seite Service-Updates in der Konsole.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Amazon-Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Klicken Sie im Navigationsbereich auf Service-Updates.
3. Unter Service updates (Service-Updates) können Sie Folgendes anzeigen:
 - Name des Service-Updates: Der eindeutige Name des Service-Updates
 - 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-OSS) Clustern das neueste Service-Update ausgeführt wird, indem Sie die folgenden Schritte ausführen:

1. Wählen Sie auf der Seite OSSRedis-Cluster einen geeigneten Cluster aus
2. Wählen Sie im Navigationsbereich Service Updates aus, um die entsprechenden Service-Updates für diesen Cluster, falls vorhanden, zu sehen.

Wenn in der Konsole eine Liste mit Service-Updates angezeigt wird, können Sie das Service-Update auswählen und auf Jetzt anwenden klicken.

Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-b...	Status	Cluster ...
elasticache-redis-6-2-6-update-202310109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 03:59:59 (UT...	Available	March 1, 2...

Wenn auf der Konsole „Keine Dienstupdates gefunden“ angezeigt wird, bedeutet dies, dass auf dem ElastiCache (Redis-OSS) Cluster bereits das neueste Service-Update installiert wurde.

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](#).

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](#)
- [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-Memcached-Caches.

Metrik	Beschreibung	Einheit
BytesUsedForCache	Die Gesamtzahl der von den in Ihrem Cache gespeicherten Daten verwendeten Bytes.	Bytes

Metrik	Beschreibung	Einheit
ElastiCacheProcessingUnits	Die Gesamtzahl der ElastiCacheProcessingUnits (ECPUs), die von den in Ihrem Cache ausgeführten Anfragen verbraucht werden	Anzahl
SuccessfulReadRequestLatency	Latenz erfolgreicher Leseanfragen.	Mikrosekunden
SuccessfulWriteRequestLatency	Latenz erfolgreicher Schreibanforderungen	Mikrosekunden
TotalCmdsCount	Gesamtzahl aller Befehle, die in Ihrem Cache ausgeführt wurden	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
NewConnections	Gesamtanzahl der Verbindungen, die in diesem Zeitraum vom Server akzeptiert worden sind.	Anzahl
CurrItems	Anzahl der Elemente im Cache.	Anzahl
NetworkBytesIn	Gesamtzahl der in den Cache übertragenen Bytes	Bytes

Metrik	Beschreibung	Einheit
NetworkBytesOut	Gesamtzahl der aus dem Cache übertragenen Bytes	Bytes
Evictions	Die Anzahl der Schlüssel, die vom Cache bereinigt wurden	Anzahl
Zurückgefordert	Die Anzahl der Schlüssel, die im Cache abgelaufen sind	Anzahl

Metriken auf Befehlsebene

ElastiCache gibt außerdem die folgenden Memcached-Metriken auf Befehlsebene aus.

Metrik	Beschreibung	Einheit
CmdGet	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
CmdSet	Die Anzahl der Set-Befehle, die der Cache empfangen hat.	Anzahl
CmdTouch	Die Anzahl der touch-Befehle, die der Cache erhalten hat.	Anzahl
GetHits	Die Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
GetMisses	Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl

Metrik	Beschreibung	Einheit
IncrHits	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrMisses	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
DecrHits	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DecrMisses	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
DeleteHits	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl

Metrik	Beschreibung	Einheit
DeleteMisses	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
TouchHits	Die Anzahl der Schlüssel, auf die zugegriffen wurde und die mit einer neuen Ablaufzeit versehen wurden.	Anzahl
TouchMisses	Die Anzahl der Schlüssel, auf die zwar zugegriffen wurde, die aber nicht gefunden werden konnten.	Anzahl
CasHits	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel gefunden wurde und die CAS-Werte übereinstimmen.	Anzahl
CasMisses	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
CasBadval	Die Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der CAS-Wert nicht mit dem gespeicherten CAS-Wert übereinstimmt.	Anzahl

Metrik	Beschreibung	Einheit
CmdFlush	Anzahl der flush-Befehle, die der Cache erhalten hat.	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 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. Im Bildschirm Ereignisse repräsentiert jede Zeile in der Liste 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-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

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
				Subnetze angegeben.
Cache-Erstellung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Der Cache <cache-name> konnte nicht erstellt werden. Das Kontingentlimit für die Erstellung eines VPC-Endpunkts wurde erreicht.
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.

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-

detail-type	Liste der Ressourcen	Kategorie	Quelle	Fehlermeldung
				name> wurden aktualisiert.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. SecurityGroups-Update fehlgeschlagen.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Konfigurationsänderung	Serverless-Cache	Eine Aktualisierung des Caches <cache-name> ist fehlgeschlagen. Die SecurityGroups-Aktualisierung ist aufgrund unzureichender Berechtigungen 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 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.

detail-type	Beschreibung	Einheit	Quelle	Fehlermeldung
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist fehlgeschlagen, da der Cache gelöscht wurde.
Cache-Aktualisierung fehlgeschlagen	Cache arn	Ausfall	Serverless-Cache	Eine Aktualisierung des Limits für den Cache <cache-name> ist aufgrund einer ungültigen Konfiguration fehlgeschlagen.

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](#)
- [Überwachung der Nutzung mithilfe von CloudWatch-Metriken](#)
- [SNSÜberwachung von ElastiCache Ereignissen durch Amazon](#)

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 für Memcached finden Sie unter [Metriken auf Host-Ebene](#).

Weitere Informationen zu Memcached-Metriken auf Knotenebene finden Sie unter [Metriken für Memcached](#).

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 von oder mithilfe von Amazon Simple Notification Service (SNS) anzeigen. AWS CLI Selbst entworfene Cluster-Ereignisse werden nicht auf Amazon EventBridge veröffentlicht.

Informationen über Ereignisse für selbst entworfene Cluster umfassen Datum und Zeit eines Ereignisses, den Quellnamen und Quelltyp sowie eine Beschreibung des Ereignisses. Sie können Ereignisse mithilfe der ElastiCache Konsole, des Befehls AWS CLI `describe-events` oder der API-Aktion ganz einfach aus dem ElastiCache Protokoll abrufen. `DescribeEvents`

ElastiCache Ereignisse anzeigen (Konsole)

Das folgende Verfahren zeigt Ereignisse mithilfe der ElastiCache Konsole an.

Um Ereignisse mit der ElastiCache Konsole anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>
2. Um eine Liste aller verfügbaren Ereignisse anzuzeigen, wählen Sie im Navigationsbereich Events (Ereignisse).
3. Im Bildschirm Ereignisse repräsentiert jede Zeile in der Liste 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.

ElastiCache Ereignisse anzeigen (AWS CLI)

Um mit dem eine Liste von ElastiCache Ereignissen zu generieren AWS CLI, verwenden Sie den Befehl `describe-events`. Mit optionalen Parametern können Sie u. a. den Typ und den Zeitrahmen der aufgelisteten Ereignisse sowie die maximale Anzahl der aufzulistenden Ereignisse steuern.

Mit dem folgenden Code werden bis zu 40 Ereignisse für selbst entworfene Cluster aufgelistet.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

Mit dem folgenden Code werden alle Ereignisse der letzten 24 Stunden (1 440 Minuten) für selbst entworfene Caches aufgelistet.

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```


Ereignisse für selbst entworfene Cluster


Dieser Abschnitt enthält die Liste der Ereignisse, die Sie für Ihre selbst entworfenen Cluster erwarten können.

Die folgenden ElastiCache Ereignisse lösen Amazon SNS 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 aufgrund unzureichender freier IP-Adressen	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Ein Cache-Knoten konnte nicht hinzugefügt werden, da nicht genug IP-Adressen verfügbar sind.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Es wurde mindestens ein Parameter des Cache-Clusters geändert.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	Die Bereitstellung eines Cache-Clusters wurde abgeschlossen und die Cache-Knoten im Cache-Cluster stehen zur Verwendung bereit.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache: CacheClusterProvisioningFailed aufgrund eines inkompatiblen Netzwerkzustands	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>Clustername</i>	Die Erweiterung des Cache-Clusters ist fehlgeschlagen.
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 festgestellt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, heruntergestuft oder nicht erreichbar ist, und hat begonnen, den Cache-Knoten zu ersetzen.</p> <div data-bbox="1068 590 1510 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-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache hat festgestellt, dass der Host, auf dem ein Cacheknoten ausgeführt wird, heruntergefahren oder nicht erreichbar ist, und hat den Austausch des Cacheknotens abgeschlossen.</p> <div data-bbox="1068 590 1507 905"><p> 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-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Mindestens ein Cache-Knoten wurde neu gestartet. Nachricht (Memcached): "Cache node %s shutdown" Dann eine zweite Nachricht: "Cache node %s restarted"
ElastiCache:CertificateRenewalComplete (Nur Redis OSS)	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 OSS)	ElastiCache:FailoverComplete : <i>mycluster</i>	Failover zu einem Replikationsknoten wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Die Anzahl der Replikate im Cluster wurde erhöht.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Der Vorgang für das Hinzufügen von Replikaten zu Ihrem Cluster hat begonnen.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster, der ersetzt werden sollte, soll nicht länger ersetzt werden.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>Für einen zu ersetzenden Knoten in Ihrem Cluster wurde eine spätere Ersetzung im neuen Fenster geplant, das in der Benachrichtigung angegeben wurde.</p> <p>Informationen darüber, welche Aktionen Sie ausführen können, finden Sie unter Ersetzen von Cache-Knoten für Memcached.</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>Informationen darüber, welche Aktionen Sie ausführen können, finden Sie unter Ersetzen von Cache-Knoten für Memcached.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Ein Cache-Knoten wurde aus dem Cache-Cluster entfernt.
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 OSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Ein Cache-Snapshot wurde erfolgreich abgeschlossen.
ElastiCache: SnapshotFailed (Nur Redis OSS)	SnapshotFailed : <i>cluster-name</i>	Ein Cache-Snapshot ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Cache-Ereignissen des Clusters. Wenn Sie den Snapshot beschreiben (siehe DescribeSnapshots), ist dessen Status failed.

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.

Wichtig

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 Memcached](#)
- [Welche Metriken sollte ich überwachen?](#)
- [Überwachung von CloudWatch-Cluster- und Knoten-Metriken](#)

Metriken auf Host-Ebene

Der AWS/ElastiCache-Namespace enthält die folgenden Metriken auf Host-Ebene für einzelne Cache-Knoten.

Weitere Informationen finden Sie auch unter:

- [Metriken für Memcached](#)

Metrik	Beschreibung	Einheit
CPUUtilization	Der Prozentsatz der CPU-Nutzung für den gesamten Host.	Prozent

Metrik	Beschreibung	Einheit
CPUCreditBalance	<p>Die Anzahl verdienter CPU-Guthaben, die eine Instance angesammelt hat, seit sie gestartet wurde. Für T2 Standard beinhaltet der CPUCreditBalance 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 CPUCreditBalance 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>	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>	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
NetworkPacketsIn	Anzahl der von der Instance auf allen Netzwerkschnittstellen empfangenen Pakete. Diese Metrik gibt das an eine einzelne Instance eingehende Netzwerkdatenvolumen an, ausgedrückt in Anzahl an Paketen.	Anzahl

Metrik	Beschreibung	Einheit
NetworkPacketsOut	Anzahl der von der Instance auf allen Netzwerkschnittstellen gesendeten Pakete. Diese Metrik gibt das von einer einzelnen Instance ausgehende Netzwerkdatenvolumen an, ausgedrückt in Anzahl an Paketen.	Anzahl
NetworkBandwidthIn AllowanceExceeded	Die Anzahl der Pakete wurde geformt, weil die eingehende aggregierte Bandbreite das Maximum für die Instance überschritten hat.	Anzahl
NetworkConntrackAllowanceExceeded	Die Anzahl der Pakete wurde geformt, 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
NetworkBandwidthOut AllowanceExceeded	Die Anzahl der Pakete wurde geformt, weil die ausgehende aggregierte Bandbreite das Maximum für die Instance überschritten hat.	Anzahl
Network Packets Per Second Allowance Exceeded	Die Anzahl der Pakete, die geformt 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
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

Metrik	Beschreibung	Einheit
SwapUsage	Größe des belegten Auslagerungsspeichers auf dem Host.	Bytes

Metriken für Memcached

Der AWS/ElastiCache-Namespace enthält die folgenden Memcache-Metriken.

Der ElastiCache Namespace AWS/enthält die folgenden Metriken, die vom Befehl Memcached stats abgeleitet sind. Jede Metrik wird zu jeder Cache-Knotenebene berechnet.

Informationen finden Sie auch unter:

- [Metriken auf Host-Ebene](#)

Metrik	Beschreibung	Einheit
BytesReadIntoMemcached	Anzahl der Byte, die vom Cache-Knoten aus dem Netzwerk gelesen wurden.	Bytes
BytesUsedForCacheItems	Anzahl der Byte, die zum Speichern von Cache-Elementen verwendet werden.	Bytes
BytesWrittenOutFromMemcached	Anzahl der Byte, die vom Cache-Knoten in das Netzwerk geschrieben wurden.	Bytes
CasBadval	Anzahl der CAS-Anforderungen (check and set), die vom Cache empfangen wurden, bei denen der CAS-Wert nicht mit dem gespeicherten CAS-Wert übereinstimmt.	Anzahl
CasHits	Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel gefunden wurde und die CAS-Werte übereinstimmen.	Anzahl

Metrik	Beschreibung	Einheit
CasMisses	Anzahl der CAS-Anforderungen, die vom Cache empfangen wurden, bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
CmdFlush	Anzahl der flush-Befehle, die der Cache erhalten hat.	Anzahl
CmdGet	Die Anzahl der get-Befehle, die der Cache erhalten hat.	Anzahl
CmdSet	Die Anzahl der Set-Befehle, die der Cache empfangen hat.	Anzahl
CurrConnections	<p>Die Anzahl der zu einem bestimmten Zeitpunkt mit dem Cache verbundenen Verbindungen. ElastiCache verwendet zwei bis drei der Verbindungen zur Überwachung des Clusters.</p> <p>Zusätzlich zu den oben genannten erstellt memcached eine Anzahl von internen Verbindungen, die dem Doppelten der für den Knotentyp verwendeten Threads entsprechen. Die Thread-Anzahl für die verschiedenen Knotentypen ist in <code>Nodetype Specific Parameters</code> der entsprechenden Parametergruppe zu sehen.</p> <p>Die Gesamtzahl der Verbindungen ist die Summe der Clientverbindungen, der zu überwachenden Verbindungen und der oben genannten internen Verbindungen.</p>	Anzahl
CurrItems	Die Anzahl der zu einem bestimmten Zeitpunkt mit dem Cache verbundenen Verbindungen.	Anzahl

Metrik	Beschreibung	Einheit
DecrHits	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DecrMisses	Die Anzahl der Dekrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
DeleteHits	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
DeleteMisses	Die Anzahl der Löschanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
Evictions	Die Anzahl nicht abgelaufener Elemente, die vom Cache bereinigt wurden, um Platz für neue Schreibvorgänge zu schaffen.	Anzahl
GetHits	Die Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel gefunden wurde.	Anzahl
GetMisses	Anzahl der get-Anforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
IncrHits	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl

Metrik	Beschreibung	Einheit
IncrMisses	Die Anzahl der Inkrementierungsanforderungen, die vom Cache empfangen wurden und bei denen der angeforderte Schlüssel nicht gefunden wurde.	Anzahl
Reclaimed	Die Anzahl abgelaufener Elemente, die vom Cache bereinigt wurden, um Platz für neue Schreibvorgänge zu schaffen.	Anzahl

Für Memcached 1.4.14 stehen die folgenden zusätzlichen Metriken zur Verfügung.

Metrik	Beschreibung	Einheit
BytesUsedForHash	Die Anzahl der Byte, die derzeit von Hash-Tabellen verwendet werden.	Bytes
CmdConfigGet	Die kumulative Anzahl an config get-Anforderungen.	Anzahl
CmdConfigSet	Die kumulative Anzahl an config set-Anforderungen.	Anzahl
CmdTouch	Die kumulative Anzahl an touch-Anforderungen.	Anzahl
CurrConfig	Die aktuelle Anzahl gespeicherter Konfigurationen.	Anzahl
EvictedUnfetched	Die Anzahl der aus dem LRU-Cache (Last Recently Used, "am längsten nicht verwendet") entfernten gültigen Elemente, auf die nie zugegriffen wurde, nachdem sie festgelegt worden waren.	Anzahl
ExpiredUnfetched	Die Anzahl der aus dem LRU-Cache wiedergewonnenen abgelaufenen Elemente,	Anzahl

Metrik	Beschreibung	Einheit
	auf die nie zugegriffen wurde, nachdem sie festgelegt worden waren.	
SlabsMoved	Die Gesamtanzahl der Slab Pages, die verschoben worden sind.	Anzahl
TouchHits	Die Anzahl der Schlüssel, auf die zugegriffen wurde und die mit einer neuen Ablaufzeit versehen wurden.	Anzahl
TouchMisses	Die Anzahl der Elemente, auf die zwar zugegriffen wurde, die aber nicht gefunden werden konnten.	Anzahl

Der ElastiCache Namespace `AWS/`enthält die folgenden berechneten Metriken auf Cache-Ebene.

Metrik	Beschreibung	Einheit
NewConnections	Die Anzahl der neuen Verbindungen, die der Cache erhalten hat. Dies wird von der memcached-Statistik <code>total_connections</code> abgeleitet, indem die Änderung in <code>total_connections</code> über einen bestimmten Zeitraum aufgezeichnet wird. Dieser Wert wird immer mindestens 1 sein, da eine Verbindung für <code>a</code> reserviert ist. ElastiCache	Anzahl
NewItems	Die Anzahl der neuen Elemente, die im Cache gespeichert wurden. Dies wird von der memcached-Statistik <code>total_items</code> abgeleitet, indem die Änderung in <code>total_items</code> über einen bestimmten Zeitraum aufgezeichnet wird.	Anzahl
UnusedMemory	Die Größe des nicht von Daten belegten Speichers. Dies wird von den Memcached-Statistiken <code>limit_maxbytes</code> und <code>bytes</code> abgeleitet.	Bytes

Metrik	Beschreibung	Einheit
	<p>t, indem bytes von limit_maxbytes subtrahiert werden.</p> <p>Da der Memcached-Overhead zusätzlich zu dem für Daten verwendeten Speicher belegt, UnusedMemory sollte dieser Wert nicht als die Menge an Speicher angesehen werden, die für zusätzliche Daten zur Verfügung steht. Möglicherweise treten Bereinigungen auch dann auf, wenn noch etwas ungenutzter Speicherplatz vorhanden ist.</p> <p>Detailliertere Informationen erhalten Sie unter Memcached item memory usage.</p>	

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](#)
- [SwapUsage](#)
- [Evictions](#)
- [CurrConnections](#)

CPUUtilization

Diese Metrik auf Hostebene wird in Prozent angegeben. Weitere Informationen finden Sie unter [Metriken auf Host-Ebene](#).

Da Memcached mit mehreren Threads arbeitet, darf diese Metrik bis zu 90 % erreichen. Wird dieser Grenzwert überschritten, erweitern Sie den Cache-Cluster, indem Sie einen größeren Cache-Knotentyp verwenden oder weitere Cache-Knoten hinzufügen.

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. In diesem Fall empfehlen wir, den Parameterwert für `ConnectionOverhead` zu erhöhen.

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.

Wird der festgelegte Grenzwert überschritten, erweitern Sie den Cluster, indem Sie einen größeren Knotentyp verwenden oder weitere Knoten hinzufügen.

CurrConnections

Dies ist eine Metrik für die Cache-Engine. Wir empfehlen Ihnen, einen eigenen Grenzwert für diese Metrik basierend auf den Anforderungen Ihrer Anwendung zu bestimmen.

Ein höherer Wert für CurrConnections kann auf ein Problem mit der Anwendung hinweisen. Untersuchen Sie das Verhalten der Anwendung, um das Problem zu beheben.

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>
```

SNSÜberwachung von ElastiCache Ereignissen durch Amazon

Wenn für einen Cluster wichtige Ereignisse eintreten, ElastiCache sendet eine Benachrichtigung an ein bestimmtes SNS Amazon-Thema. Zu den wichtigen Ereignissen zählen beispielsweise das fehlgeschlagene Hinzufügen eines Knotens, das erfolgreiche Hinzufügen eines Knotens und die Änderung einer Sicherheitsgruppe. Durch die Überwachung wichtiger Schlüsselereignisse können Sie den aktuellen Status Ihrer Cluster erfahren und, je nach Ereignis, Korrekturen vornehmen.

Themen

- [ElastiCache SNSAmazon-Benachrichtigungen verwalten](#)
- [Anzeigen von ElastiCache-Ereignissen](#)
- [Ereignisbenachrichtigungen und Amazon SNS](#)

ElastiCache SNSAmazon-Benachrichtigungen verwalten

Mit Amazon Simple Notification Service (AmazonSNS) können Sie konfigurieren ElastiCache , dass Benachrichtigungen für wichtige Cluster-Ereignisse gesendet werden. In diesen Beispielen konfigurieren Sie einen Cluster mit dem Amazon-Ressourcennamen (ARN) eines SNS Amazon-Themas für den Empfang von Benachrichtigungen.

Note

- In diesem Thema wird davon ausgegangen, dass Sie sich bei Amazon angemeldet SNS und ein SNS Amazon-Thema eingerichtet und abonniert haben. Informationen dazu finden Sie im [Entwicklerhandbuch zu Amazon Simple Notification Service](#).
- Standardmäßig `API modify-replication-group` betrifft das alle Gruppen in einer Region und nicht nur die aktuell angegebene Gruppe. Wenn Sie eine bestimmte Gruppe in einer Region anders konfigurieren möchten als die anderen Gruppen, können Sie die `--notification-topic-arn` Option verwenden, um ein separates Thema für diese Gruppe zu erstellen.

Ein SNS Amazon-Thema hinzufügen

In den folgenden Abschnitten erfahren Sie, wie Sie mithilfe der AWS Konsole, der oder der AWS CLI ein SNS Amazon-Thema hinzufügen ElastiCache API.

Ein SNS Amazon-Thema hinzufügen (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie ein SNS Amazon-Thema für einen Cluster hinzufügen.

Note

Dieser Prozess kann auch verwendet werden, um das SNS Amazon-Thema zu ändern.

Um ein SNS Amazon-Thema für einen Cluster hinzuzufügen oder zu ändern (Konsole)

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Wählen Sie unter Cluster den Cluster aus, für den Sie ein SNS Amazon-Thema hinzufügen oder ändern möchtenARN.
3. Wählen Sie Ändern aus.
4. Wählen Sie unter Cluster modifizieren unter Thema für SNS Benachrichtigung das SNS Thema aus, das Sie hinzufügen möchten, oder wählen Sie Manuelle ARN Eingabe und geben Sie das ARN SNS Amazon-Thema ein.
5. Wählen Sie Ändern aus.

Ein SNS Amazon-Thema hinzufügen (AWS CLI)

Verwenden Sie den AWS CLI Befehl, um ein SNS Amazon-Thema für einen Cluster hinzuzufügen oder zu ändern`modify-cache-cluster`.

Das folgende Codebeispiel fügt ein SNS Amazon-Thema `arn` zu `my-cluster` hinzu.

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-arn arn:aws:sns:us-  
west-2:123456789xxx:ElastiCacheNotifications
```

Für Windows:

```
aws elasticache modify-cache-cluster ^
  --cache-cluster-id my-cluster ^
  --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

Weitere Informationen finden Sie unter [modify-cache-cluster](#).

Ein SNS Amazon-Thema hinzufügen (ElastiCache API)

Um ein SNS Amazon-Thema für einen Cluster hinzuzufügen oder zu ändern, rufen Sie die `ModifyCacheCluster` Aktion mit den folgenden Parametern auf:

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/
  ?Action=ModifyCacheCluster
  &ApplyImmediately=false
  &CacheClusterId=my-cluster
  &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-
west-2%3A565419523791%3AElastiCacheNotifications
  &Version=2014-12-01
  &SignatureVersion=4
  &SignatureMethod=HmacSHA256
  &Timestamp=20141201T220302Z
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256
  &X-Amz-Date=20141201T220302Z
  &X-Amz-SignedHeaders=Host
  &X-Amz-Expires=20141201T220302Z
  &X-Amz-Credential=<credential>
  &X-Amz-Signature=<signature>
```

Weitere Informationen finden Sie unter [ModifyCacheCluster](#).

SNSAmazon-Benachrichtigungen aktivieren und deaktivieren

Sie können Benachrichtigungen für einen Cluster aktivieren oder deaktivieren. Die folgenden Verfahren zeigen Ihnen, wie Sie SNS Amazon-Benachrichtigungen deaktivieren können.

SNSAmazon-Benachrichtigungen aktivieren und deaktivieren (Konsole)

Um SNS Amazon-Benachrichtigungen zu deaktivieren, verwenden Sie den AWS Management Console

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.
2. Um eine Liste der Cluster anzuzeigen, die Memcached ausführen, wählen Sie im Navigationsbereich Memcached.
3. Aktivieren Sie das Kontrollkästchen links neben dem Cluster, für den die Benachrichtigung geändert werden soll.
4. Wählen Sie Ändern aus.
5. Wählen Sie unter „Cluster modifizieren“ unter „Thema für SNS Benachrichtigungen“ die Option Benachrichtigungen deaktivieren aus.
6. Wählen Sie Ändern aus.

SNSAmazon-Benachrichtigungen aktivieren und deaktivieren (AWS CLI)

Verwenden Sie den Befehl `modify-cache-cluster` mit den folgenden Parametern, um SNS Amazon-Benachrichtigungen zu deaktivieren:

Für Linux, macOS oder Unix:

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-status inactive
```

Für Windows:

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-status inactive
```

SNSAmazon-Benachrichtigungen aktivieren und deaktivieren () ElastiCache API

Um SNS Amazon-Benachrichtigungen zu deaktivieren, rufen Sie die `ModifyCacheCluster` Aktion mit den folgenden Parametern auf:

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

Diese Aktion führt zu folgender oder einer ähnlichen Ausgabe:

Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&ApplyImmediately=false  
&CacheClusterId=my-cluster  
&NotificationTopicStatus=inactive  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

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, dem das Amazon SNS SNS-Thema gehört, muss dasselbe Konto sein, das den Cache-Cluster besitzt, auf dem 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 in der ElastiCache Konsole wird jedoch als inaktiv angezeigt, wodurch das Thema effektiv vom Cluster getrennt wird, wenn Nachrichten an das ElastiCache Thema weitergeleitet werden.

- Das Amazon SNS SNS-Thema muss sich in derselben Region wie der ElastiCache Cluster befinden.


ElastiCache Ereignisse


Die folgenden ElastiCache Ereignisse lösen Amazon SNS 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 aufgrund unzureichender freier IP-Adressen	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	Ein Cache-Knoten konnte nicht hinzugefügt werden, da nicht genug IP-Adressen verfügbar sind.
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	Es wurde mindestens ein Parameter des Cache-Clusters geändert.
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioning	Die Bereitstellung eines Cache-Clusters wurde

Ereignisname	Fehlermeldung	Beschreibung
	Complete <i>cluster-name-0001-005</i>	abgeschlossen und die Cache-Knoten im Cache-Cluster stehen zur Verwendung bereit.
ElastiCache: CacheClusterProvisioningFailed aufgrund eines inkompatiblen Netzwerkzustands	ElastiCache:CacheClusterProvisioning Failed : <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>Clustername</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 festgestellt, dass der Host, auf dem ein Cache-Knoten ausgeführt wird, heruntergestuft oder nicht erreichbar ist, und hat begonnen, den Cache-Knoten zu ersetzen.</p> <div data-bbox="1068 590 1507 905"><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-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache hat festgestellt, dass der Host, auf dem ein Cacheknoten ausgeführt wird, heruntergefahren oder nicht erreichbar ist, und hat den Austausch des Cacheknotens abgeschlossen.</p> <div data-bbox="1068 590 1507 905" 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-Client-Bibliotheken verwenden den Cache-Knoten möglicherweise auch dann nicht mehr, wenn der Cache-Knoten ersetzt wurde. In diesem Fall sollte die Anwendung die Serverliste aktualisieren, wenn dieses Ereignis eintritt.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	Mindestens ein Cache-Knoten wurde neu gestartet. Nachricht (Memcached): "Cache node %s shutdown" Dann eine zweite Nachricht: "Cache node %s restarted"
ElastiCache:CertificateRenewalComplete (Nur Redis OSS)	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 OSS)	ElastiCache:FailoverComplete : <i>mycluster</i>	Failover zu einem Replikationsknoten wurde erfolgreich abgeschlossen.
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	Die Anzahl der Replikate im Cluster wurde erhöht.

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	Der Vorgang für das Hinzufügen von Replikaten zu Ihrem Cluster hat begonnen.
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	Ein Knoten in Ihrem Cluster, der ersetzt werden sollte, soll nicht länger ersetzt werden.
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>Für einen zu ersetzenden Knoten in Ihrem Cluster wurde eine spätere Ersetzung im neuen Fenster geplant, das in der Benachrichtigung angegeben wurde.</p> <p>Informationen darüber, welche Aktionen Sie ausführen können, finden Sie unter Ersetzen von Cache-Knoten für Memcached.</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>Informationen darüber, welche Aktionen Sie ausführen können, finden Sie unter Ersetzen von Cache-Knoten für Memcached.</p>

Ereignisname	Fehlermeldung	Beschreibung
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	Ein Cache-Knoten wurde aus dem Cache-Cluster entfernt.
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 OSS)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	Ein Cache-Snapshot wurde erfolgreich abgeschlossen.
ElastiCache: SnapshotFailed (Nur Redis OSS)	SnapshotFailed : <i>cluster-name</i>	Ein Cache-Snapshot ist fehlgeschlagen. Weitere Informationen zur Ursache finden Sie in den Cache-Ereignissen des Clusters. Wenn Sie den Snapshot beschreiben (siehe 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"
}
```

Kontingente für ElastiCache

Ihr AWS Konto verfügt über Standardkontingente, die früher als Limits bezeichnet wurden, für jeden AWS Dienst. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Um die Kontingente für anzuzeigen ElastiCache, öffnen Sie die [Konsole Service Quotas](#). Wählen Sie im Navigationsbereich AWS Dienste aus und wählen Sie ElastiCache.

Informationen zur Erhöhung eines Kontingents finden Sie unter [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch zu Service Quotas. Wenn das Kontingent unter Service Quotas noch nicht in verfügbar ist, verwenden Sie das [Formular zur Erhöhung des Service-Limits](#).

Für Ihr AWS Konto gelten die folgenden Kontingente für ElastiCache.

Ressource	Standard
Serverless-Caches pro Region	40
Knoten pro Region	300
Knoten pro Cluster	60
Parametergruppen pro Region	300
Sicherheitsgruppen pro Region	50
Subnetzgruppen pro Region	300
Subnetze pro Subnetzgruppe	20

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](#)
- [ElastiCache Amazon-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
```

ElastiCache Amazon-Fehlermeldungen

Die folgenden Fehlermeldungen werden von Amazon zurückgegeben ElastiCache. Möglicherweise erhalten Sie weitere Fehlermeldungen, die von anderen AWS Diensten oder von ElastiCache Memcached zurückgegeben werden. Beschreibungen von Fehlermeldungen aus anderen Quellen als ElastiCache finden Sie in der Dokumentation der Quelle, die die Fehlermeldung generiert hat.

- [Cluster node quota exceeded](#)
- [Customer's node quota exceeded](#)
- [Insufficient cache cluster capacity](#)

Fehlermeldung: Das Cluster-Knoten-Kontingent ist erschöpft. Jeder Cluster in dieser Region kann maximal %n Knoten enthalten.

Ursache: Sie haben versucht, einen Cluster zu erstellen oder zu ändern, der dadurch mehr als %n Knoten enthalten würde.

Lösung: Ändern Sie die Anforderung dahingehend, dass der Cluster nicht mehr als %n Knoten enthält. Oder, wenn Sie mehr als %n Knoten benötigen, stellen Sie Ihre Anfrage über das [Amazon ElastiCache Node-Anfrageformular](#).

Weitere Informationen finden Sie unter [ElastiCache Amazon-Limits](#) unter Allgemeine Amazon Web Services-Referenz.

Fehlermeldungen: Kundenknotenquote überschritten. Sie können maximal %n Knoten in dieser Region haben. Oder: Sie haben Ihr Kontingent an %s-Knoten in dieser Region bereits erreicht.

Ursache: Sie haben versucht, einen Cluster zu erstellen oder zu ändern, und Ihr Konto würde dadurch mehr als %n Knoten in allen Clustern in dieser Region enthalten.

Lösung: Ändern Sie die Anforderung dahingehend, dass die Gesamtanzahl der Knoten in der Region für alle Cluster in diesem Konto nicht mehr als &n beträgt. Oder, wenn Sie mehr als %n Knoten benötigen, stellen Sie Ihre Anfrage über das [Amazon ElastiCache Node-Anfrageformular](#).

Weitere Informationen finden Sie unter [ElastiCache Amazon-Limits](#) unter Allgemeine Amazon Web Services-Referenz.

Fehlermeldungen: 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 \(Memcached\)](#).

Benachrichtigungen

In diesem Thema werden ElastiCache Benachrichtigungen behandelt, die Sie interessieren könnten. Eine Benachrichtigung ist eine Situation oder ein Ereignis, die bzw. das in der Regel temporär ist und nur solange andauert, bis eine Lösung gefunden und implementiert wurde. Benachrichtigungen haben normalerweise ein Anfangsdatum und ein Lösungsdatum, nach dem die Benachrichtigung nicht mehr relevant ist. Sie entscheiden selbst, welche Benachrichtigungen für Sie relevant sind. Wir empfehlen eine Implementierungsrichtlinie, durch die sich die Leistung Ihres Clusters verbessern lässt.

Benachrichtigungen kündigen keine neuen oder verbesserten ElastiCache Features oder Funktionen an.

Allgemeine ElastiCache Benachrichtigungen

Derzeit gibt es keine ausstehenden ElastiCache Benachrichtigungen, die nicht maschinenspezifisch sind.


ElastiCache (Memcached) Benachrichtigungen

Die folgenden ElastiCache Benachrichtigungen sind spezifisch für die Memcached-Engine.

ElastiCache (Memcached) -spezifische Benachrichtigungen

- [Warnung: Memcached-LRU-Crawler verursacht Segmentierungsfehler](#)

Warnung: Memcached-LRU-Crawler verursacht Segmentierungsfehler

 Datum der Warnung: 28. Februar 2017

Unter bestimmten Umständen kann ein Cluster durch einen Segmentierungsfehler im Memcached-LRU-Crawler instabil werden. Dieses Problem tritt bereits seit einiger Zeit in der Memcached-Engine auf. Es trat erstmals in Memcached 1.4.33 auf, wenn der LRU-Crawler standardmäßig aktiviert wurde.

Wenn dieses Problem bei Ihnen auftritt, empfehlen wir, den LRU-Crawler zu deaktivieren, bis das Problem behoben wurde. Führen Sie dazu den Befehl `lru_crawler disable` auf der Befehlszeile aus oder ändern Sie den Parameterwert `lru_crawler` (empfohlen).

Lösungsdatum:

Auflösung

der Dokumentation

- APIVersion: 2015-02-02
- Letzte Aktualisierung der Dokumentation: 27. November 2023

In der folgenden Tabelle werden wichtige Änderungen in den einzelnen Versionen des ElastiCache (Memcached) User Guide (Redis) User Guide Guide nach März 2018 beschrieben. Wenn Sie über Aktualisierungen dieser Dokumentation informiert werden möchten, können Sie den Feed abonnieren. [RSS](#)

Aktuelle ElastiCache (Memcached) Updates

Änderung	Beschreibung	Datum
Support für ElastiCache (Memcached) 1.6.22	ElastiCache (Memcached) unterstützt jetzt Memcached 1.6.22. Es enthält kumulative Fehlerkorrekturen der Version 1.6.18 . Weitere Informationen finden Sie unter Memcached Version 1.6.22 .	10. Januar 2024
ElastiCache (Memcached) unterstützt jetzt die Erstellung von serverlosen Caches	Sie können jetzt Serverless-Caches erstellen, die das Cache-Management vereinfachen und sofort skalieren, um die anspruchsvollsten Anwendungen zu unterstützen. Weitere Informationen finden Sie unter Auswählen aus den Bereitstellungsoptionen . Im Rahmen dieser Funktion wurden neue Berechtigungen hinzugefügt, um die Zuordnung von serverlosen ElastiCac	8. November 2023

heServiceRolePolicy Caches AmazonElastiCacheFullAccess zu verwalteten Endpunkten zu ermöglichen. VPC Darüber hinaus wurden Berechtigungen hinzugefügt, um ein überarbeitetes Konsolenerlebnis mithilfe der Richtlinie AmazonElastiCacheFullAccess zu unterstützen.

[Support für ElastiCache \(Memcached\) 1.6.17](#)

ElastiCache (Memcached) unterstützt jetzt Memcached 1.6.17. Es enthält kumulative Bugfixes aus Version [Memcached 1.6.12](#). Weitere Informationen finden Sie unter [Memcached Version 1.6.17](#).

18. Januar 2023

[ElastiCache \(Memcached\) unterstützt jetzt IPv6](#)

ElastiCache unterstützt die Internetprotokoll-Versionen 4 und 6 (IPv4 und IPv6), sodass Sie Ihren Cluster so konfigurieren können, dass er nur IPv4 Verbindungen, nur Verbindungen oder beides IPv4 und IPv6 IPv6 Verbindungen (Dual-Stack) akzeptiert. [IPv6 wird für Workloads unterstützt, die die Memcached Engine ab Version 1.6.6 auf allen auf dem Nitro-System basierenden Instanzen verwenden.](#) Für den Zugriff über fallen keine zusätzlichen Gebühren an. ElastiCache IPv6 Weitere Informationen zu Netzwerktypen finden Sie unter [Auswählen eines Netzwerktyps](#).

07. November 2022

[ElastiCache \(Memcached\) unterstützt jetzt die Verschlüsselung bei der Übertragung](#)

Die Verschlüsselung während der Übertragung ist eine optionale Funktion, mit der Sie die Sicherheit Ihrer Daten an ihrer verwundbarsten Stelle verbessern können – bei der Übertragung von einem Speicherort an einen anderen. Sie wird in Memcached-Versionen 1.6.12 und höher unterstützt. Weitere Informationen finden Sie unter [Verschlüsselung bei der ElastiCache Übertragung](#) (). TLS

26. Mai 2022

[ElastiCache unterstützt jetzt PrivateLink](#)

AWS PrivateLink ermöglicht Ihnen den privaten Zugriff auf ElastiCache API Vorgänge ohne Internet-Gateway, NAT Gerät, VPN Verbindung oder AWS Direct Connect-Verbindung. Weitere Informationen finden Sie unter [Amazon ElastiCache API und VPC Schnittstellenendpunkte \(AWS PrivateLink\)](#) für Redis OSS oder [Amazon ElastiCache API und VPC Schnittstellenendpunkte \(AWS PrivateLink\)](#) für Memcached.

24. Januar 2022

[Support für die Markierung von Ressourcen und Bedingungsschlüsseln](#)

ElastiCache unterstützt jetzt Tagging, um Sie bei der Verwaltung Ihrer Cluster und anderer Ressourcen zu unterstützen. ElastiCache Weitere Informationen finden Sie unter [Taggen Ihrer ElastiCache Ressource](#). ElastiCache führt auch Unterstützung für Bedingungsschlüssel ein. Sie können Bedingungen angeben, die bestimmen, wie eine IAM Richtlinie wirksam wird. Weitere Informationen finden Sie unter [Verwenden von Bedingungsschlüssel](#).

7. April 2021

[Support für Memcached 1.6.6.](#)

ElastiCache (Memcached) unterstützt jetzt Memcached 1.6.6. Sie enthält kumulative Fehlerkorrekturen der Version Memcached 1.5.16. Weitere Informationen finden Sie unter [Memcached Version 1.6.6](#).

12. November 2020

[ElastiCache ist jetzt auf AWS Outposts verfügbar](#)

[AWS Outposts](#) bieten native AWS Dienste, Infrastrukturen und Betriebsmodelle für praktisch jedes Rechenzentrum, jeden Colocation-Bereich oder jede lokale Einrichtung. Sie können ElastiCache auf Outposts bereitstellen, um Cache vor Ort einzurichten, zu betreiben und zu verwenden, genau wie in der Cloud. Weitere Informationen finden Sie unter [Outposts for Redis verwenden OSS oder Outposts for Memcached verwenden](#).

8. Oktober 2020

[ElastiCache unterstützt jetzt Local Zones](#)

Eine lokale Zone ist eine Erweiterung einer AWS Region, die sich geografisch in der Nähe Ihrer Benutzer befindet. Sie können jede virtuelle private Cloud (VPC) von einer übergeordneten AWS Region auf Local Zones erweitern, indem Sie ein neues Subnetz erstellen und es einer lokalen Zone zuweisen. Weitere Informationen finden Sie unter [Local Zones nutzen](#).

25. September 2020

[ElastiCache unterstützt jetzt Berechtigungen auf Ressourcenebene](#)

Sie können jetzt den Umfang der Benutzerberechtigungen einschränken, indem Sie ElastiCache Ressourcen in einer AWS Identity and Access Management (IAM) - Richtlinie angeben. Weitere Informationen finden Sie unter [Berechtigungen auf Ressourcenebene](#).

12. August 2020

[ElastiCache unterstützt jetzt die automatische Aktualisierung von Clustern ElastiCache](#)

Amazon unterstützt ElastiCache jetzt die automatische Aktualisierung von ElastiCache Clustern, nachdem das „empfohlene Verfallsdatum“ des Service-Updates abgelaufen ist. ElastiCache verwendet Ihr Wartungsfenster, um die automatische Aktualisierung der entsprechenden Cluster zu planen. Weitere Informationen finden Sie unter [Self-Service-Updates](#).

13. Mai 2020

[Amazon unterstützt ElastiCache jetzt T3-Standard-Cache-Knoten](#)

Sie können jetzt die Allzweck-Burstable-T3-Standard-Cache-Knoten der nächsten Generation in Amazon starten. ElastiCache EC2Die T3-Standard-Instances von Amazon bieten ein grundlegendes CPU Leistungsniveau mit der Möglichkeit, die CPU Nutzung jederzeit zu beschleunigen, bis das aufgelaufene Guthaben aufgebraucht ist. Weitere Informationen finden Sie unter [Unterstützte Knotentypen](#).

12. November 2019

[ElastiCache \(Memcached\) 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. Das minimiert Service-Unterbrechungen, insbesondere zu Zeiten starker betrieblicher Auslastung, und unterstützt die Erfüllung von Compliance-Anforderungen mit Sicherheitsupdates. Weitere Informationen finden Sie unter [Self-Service-Updates in Amazon ElastiCache](#).

9. Oktober 2019

[Support für ElastiCache \(Memcached\) 1.5.16](#)

ElastiCache (Memcached) unterstützt jetzt Memcached 1.5.16. Es sind kumulative Fehlerbehebungen von den Versionen [Memcached 1.5.14](#) und [Memcached 1.5.15](#) enthalten. Weitere Informationen finden Sie unter [Memcached Version 1.5.16](#).

6. September 2019

[ElastiCache Standardangebote für Reserved Instances: Teilweise Vorauszahlung, vollständige Vorauszahlung und keine Vorauszahlung.](#)

Reserved Instances bieten Ihnen die Flexibilität, je nach ElastiCache Instance-Typ und AWS Region eine Amazon-Instance für einen Zeitraum von einem oder drei Jahren zu reservieren. Weitere Informationen finden Sie unter [Verwalten der Kosten mit reservierten Knoten](#).

18. Januar 2019

[Support für ElastiCache \(Memcached\) 1.5.10](#)

ElastiCache (Memcached) unterstützt jetzt Memcached 1.5.10. Die Parameter „no_modern“ und „inline-ascii“ werden unterstützt. Weitere Informationen finden Sie unter [Memcached Version 1.5.10](#).

14. November 2018

Neu strukturiertes Benutzerhandbuch

Das einheitliche ElastiCache Benutzerhandbuch ist jetzt neu strukturiert, sodass es separate Benutzerhandbücher für Redis ((Redis) User Guide) und für Memcached OSS ElastiCache ((MemcachedOSS) User Guide) gibt. ElastiCache Die Dokumentationsstruktur im Abschnitt AWS CLI Command Reference : Elasticache und Amazon ElastiCache API Reference bleibt unverändert.

20. April 2018

In der folgenden Tabelle werden die wichtigen Änderungen am Benutzerhandbuch ElastiCache (Memcached) ElastiCache vor März 2018 beschrieben.

Änderung	Beschreibung	Änderungsdatum
Support für die Region Asien-Pazifik (Osaka – regional).	<p>ElastiCache Unterstützung für die Region Asien-Pazifik (lokal in Osaka) hinzugefügt. Die Region Asien-Pazifik (Osaka) unterstützt derzeit eine einzige Availability Zone und ist nur auf Einladung zugänglich. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> • Unterstützte Regionen • Unterstützte Cache-Knotentypen 	12. Februar 2018
Support für die Europa (Paris).	<p>ElastiCache Unterstützung für die Region EU (Paris) hinzugefügt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> • Unterstützte Regionen 	18. Dezember 2017

Änderung	Beschreibung	Änderungsdatum
	<ul style="list-style-type: none">• Unterstützte Cache-Knotentypen	
Support für die Region China (Ningxia)	<p>Amazon ElastiCache hat Unterstützung für die Region China (Ningxia) hinzugefügt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Unterstützte Regionen• Unterstützte Cache-Knotentypen	11. Dezember 2017
Unterstützung für dienstverknüpfte Rollen	<p>In dieser Version ElastiCache wurde Unterstützung für Service Linked Roles (SLR) hinzugefügt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• 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 hat Unterstützung für R4-Knotentypen in allen AWS Regionen hinzugefügt, die von ElastiCache unterstützt werden. R4-Knotentypen sind als On-Demand-Knoten oder als reservierte Cache-Knoten erhältlich. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Unterstützte Cache-Knotentypen• Spezifische Parameter des Memcached-Knotentyps	20. November 2017

Änderung	Beschreibung	Änderungsdatum
Thema der Verbindungsmuster	<p>ElastiCache Die Dokumentation fügt ein Thema hinzu, das verschiedene Muster für den Zugriff auf einen ElastiCache Cluster in einem Amazon behandeltVPC.</p> <p>Weitere Informationen finden Sie Zugriffsmuster für den Zugriff auf einen ElastiCache Cache in einer Amazon VPC im ElastiCache Benutzerhandbuch.</p>	24. April 2017
Support für Memcached 1.4.34	<p>ElastiCache fügt Unterstützung für Memcached Version 1.4.34 hinzu, die eine Reihe von Korrekturen für frühere Memcached-Versionen beinhaltet.</p> <p>Weitere Informationen finden Sie in den Versionshinweisen zu Memcached 1.4.34 auf Memcached am. GitHub</p>	10. April 2017
Unterstützung für Memcached 1.4.33	<p>ElastiCache fügt Unterstützung für Memcached Version 1.4.33 hinzu. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Memcached-Version 1.4.33• In Memcached 1.4.33 hinzugefügte Parameter	20. Dezember 2016
Unterstützung für die Region EU (London)	<p>ElastiCache fügt Unterstützung für die Region EU (London) hinzu. Derzeit werden nur die Knotentypen T2 und M4 unterstützt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Unterstützte Regionen• Unterstützte Cache-Knotentypen	13. Dezember 2016

Änderung	Beschreibung	Änderungsdatum
Unterstützung für die Region Kanada (Montreal)	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: <ul style="list-style-type: none">• Unterstützte Regionen• Unterstützte Cache-Knotentypen	8. Dezember 2016
Unterstützung für Knotentypen M4 und R3	ElastiCache fügt Unterstützung für die Knotentypen R3 und M4 in der Region Südamerika (São Paulo) und die Knotentypen M4 in der Region China (Peking) hinzu. Weitere Informationen finden Sie hier: <ul style="list-style-type: none">• Unterstützte Regionen• Unterstützte Cache-Knotentypen	1. November 2016
Unterstützung für die Region USA Ost 2 (Ohio)	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: <ul style="list-style-type: none">• Unterstützte Regionen• Unterstützte Cache-Knotentypen	17. Oktober 2016
Unterstützung für Knotentyp M4	ElastiCache fügt Unterstützung für die M4-Familie von Knotentypen in den meisten AWS Regionen hinzu, die von unterstützt werden. ElastiCache M4-Knotentypen sind als On-Demand-Knoten oder als reservierte Cache-Knoten erhältlich. Weitere Informationen finden Sie hier: <ul style="list-style-type: none">• Unterstützte Cache-Knotentypen• Spezifische Parameter des Memcached-Knotentyps	3. August 2016

Änderung	Beschreibung	Änderungsdatum
Unterstützung der Region Mumbai	<p>ElastiCache fügt Unterstützung für die Region Asien-Pazifik (Mumbai) hinzu. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none"> • Unterstützte Cache-Knotentypen • Spezifische Parameter des Memcached-Knotentyps 	27. Juni 2016
Unterstützung für R3-Knotentypen	<p>ElastiCache fügt Unterstützung für R3-Knotentypen in den Regionen China (Peking) und Südamerika (São Paulo) hinzu. Weitere Informationen finden Sie unter Unterstützte Cache-Knotentypen.</p>	16. März 2016
Zugreifen ElastiCache mit einer Lambda-Funktion	<p>Es wurde ein Tutorial zur Konfiguration einer Lambda-Funktion für den Zugriff ElastiCache in einem Amazon VPC hinzugefügt. Weitere Informationen finden Sie unter ElastiCache Tutorials und Videos.</p>	12. Februar 2016
Support für die Region Asien-Pazifik (Seoul)	<p>ElastiCache fügt Unterstützung für die Region Asien-Pazifik (Seoul) (ap-northeast-2) mit den Knotentypen t2, m3 und r3 hinzu.</p>	6. Januar 2016
Unterstützung für Memcached 1.4.28	<p>ElastiCache fügt Unterstützung für Memcached Version 1.4.24 und Memcached-Verbesserungen seit Version 1.4.14 hinzu. Diese Version bietet Unterstützung für die Cache-Verwaltung (LRU), die zuletzt als Hintergrundaufgabe verwendet wurde, die Wahl von Jenkins oder Murmur3 als Hash-Algorithmus, neue Befehle und verschiedene Bugfixes. Weitere Informationen finden Sie unter den Memcached-Versionshinweisen.</p>	27. August 2015

Änderung	Beschreibung	Änderungsdatum
Support für Memcached Auto Discovery mit 5.6 PHP	Diese Version von Amazon ElastiCache bietet Unterstützung für den Memcached Auto Discovery -Client für PHP Version 5.6. Weitere Informationen finden Sie unter Kompilieren des Quellcodes für den ElastiCache-Cluster-Client für PHP .	29. Juli 2015
Neues Thema: Zugriff ElastiCache von außen AWS	Es wurde ein neues Thema zum Zugriff auf ElastiCache Ressourcen von außen hinzugefügt AWS. Weitere Informationen finden Sie unter Zugriff ElastiCache von außen AWS .	9. Juli 2015
Nachrichten über Knotenersatz wurden hinzugefügt	ElastiCache fügt drei Meldungen hinzu, die sich auf den geplanten Knotenaustausch beziehen: ElastiCache: NodeReplacementScheduled, ElastiCache: NodeReplacementRescheduled, und ElastiCache: NodeReplacementCanceled. Weitere Informationen und Maßnahmen, die Sie ergreifen können, wenn der Austausch eines Knotens geplant ist, finden Sie unter Ereignisbenachrichtigungen und Amazon SNS .	11. Juni 2015
Unterstützung von Kostenzuordnungs-Tags	ElastiCache fügt Unterstützung für Kostenzuordnungs-Tags hinzu. Weitere Informationen finden Sie unter Überwachung von Kosten mit Kostenzuordnungs-Tags .	9. Februar 2015
Support für die AWS GovCloud Region (US-West)	ElastiCache fügt Unterstützung für die Region AWS GovCloud (US-West) (us-gov-west-1) hinzu.	29. Januar 2015
Support für die Region Europa (Frankfurt)	ElastiCache fügt Unterstützung für die Region Europa (Frankfurt) (eu-central-1) hinzu.	19. Januar 2015

Änderung	Beschreibung	Änderungsdatum
AWS CloudTrail Protokollierung von Anrufen wird unterstützt API	ElastiCache fügt Unterstützung für die Verwendung AWS CloudTrail zum Protokollieren aller ElastiCache API Anrufe hinzu. Weitere Informationen finden Sie unter Protokollieren von Amazon-ElastiCache-API-Aufrufen mithilfe von AWS CloudTrail .	15. September 2014
Es werden neue Instance-Größen unterstützt	ElastiCache fügt Unterstützung für zusätzliche General Purpose (T2) -Instances hinzu. Weitere Informationen finden Sie unter Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen .	11. September 2014
Flexibler Knotenersatz für Memcached wird unterstützt	ElastiCache fügt Unterstützung für die Erstellung von Memcached-Knoten in mehreren Availability Zones hinzu.	23. Juli 2014
Es werden neue Instance-Größen unterstützt	ElastiCache fügt Unterstützung für zusätzliche General Purpose (M3) -Instances und Memory Optimized (R3) -Instances hinzu. Weitere Informationen finden Sie unter Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen .	1. Juli 2014
PHPauto Erkennung	Unterstützung für PHP Version 5.5 Auto Discovery hinzugefügt.	13. Mai 2014
Support für die standardmäßige Amazon Virtual Private Cloud (VPC)	In dieser Version ElastiCache ist es vollständig in Amazon Virtual Private Cloud (VPC) integriert. Für Neukunden werden Cache-Cluster VPC standardmäßig in einem Amazon erstellt. Weitere Informationen finden Sie unter Amazon VPCs und ElastiCache Sicherheit .	8. Januar 2013

Änderung	Beschreibung	Änderungsdatum
PHP-Unterstützung für die auto Erkennung von Cache-Knoten	Die ursprüngliche Version von Auto Discovery für Cache-Knoten bot Unterstützung für Java-Programme. In dieser Version wird ElastiCache die auto Erkennung von Cache-Knoten unterstützt PHP.	2. Januar 2013
Support für Amazon Virtual Private Cloud (VPC)	In dieser Version können ElastiCache Cluster in Amazon Virtual Private Cloud (VPC) gestartet werden. Standardmäßig werden Cache-Cluster für neue Kunden VPC automatisch in Amazon erstellt. Bestandskunden können in ihrem eigenen Tempo zu Amazon VPC migrieren. Weitere Informationen finden Sie unter Amazon VPCs und ElastiCache Sicherheit .	20. Dezember 2012
Auto Discovery für Cache-Knoten und neue Cache-Engine-Version	<p>ElastiCache bietet die auto von Cacheknoten, d. h. die Fähigkeit von Client-Programmen, automatisch alle Cache-Knoten in einem Cluster zu ermitteln und Verbindungen zu all diesen Knoten zu initiieren und aufrechtzuerhalten.</p> <p>Diese Version bietet außerdem eine neue Cache-Engine-Version, Memcached 1.4.14. Diese neue Cache-Engine bietet erweiterte Slab-Ausgleichsfunktionen, entscheidende Leistungs- und Skalierungsverbesserungen sowie mehrere Fehlerkorrekturen. Es gibt verschiedene neue Cache-Parameter, die konfiguriert werden können. Weitere Informationen finden Sie unter Konfiguration von Engine-Parametern unter Verwendung von Parametergruppen.</p>	28. November 2012
Neue Cache-Knotentypen	Diese Version stellt vier neue Cache-Knotentypen bereit.	13. November 2012

Änderung	Beschreibung	Änderungsdatum
Reservierte Cache-Knoten	Diese Version bietet nun Unterstützung für reservierte Cache-Knoten.	5. April 2012
Neues Handbuch	Dies ist die erste Version des ElastiCache Amazon-Benutzerhandbuchs.	22. August 2011

AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.