
Amazon ElastiCache

ElastiCache for Memcached 사용 설명서

API 버전 2015-02-02



Amazon ElastiCache: ElastiCache for Memcached 사용 설명서

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

ElastiCache for Memcached란 무엇입니까?	1
일반적인 ElastiCache 사용 사례 및 ElastiCache 활용 방법	1
인 메모리 데이터 스토어	2
ElastiCache 고객 추천사	3
ElastiCache for Memcached 리소스	3
ElastiCache for Memcached 구성 요소 및 기능	4
노드	4
클러스터	5
AWS 리전 및 가용 영역	6
엔드포인트	6
파라미터 그룹	7
보안	7
보안 그룹	7
서브넷 그룹	7
이벤트	8
구현 관리용 도구	8
AWS Management 콘솔 사용	8
AWS CLI 사용	8
AWS SDK 사용	8
ElastiCache API 사용	8
다음 사항도 참조하세요.	8
Memcached와 Redis 비교	10
ElastiCache for Memcached 시작하기	13
설정	13
AWS 계정 생성	13
권한 설정(신규 ElastiCache 사용자에게만 해당)	13
Memcached 클러스터 배포	14
클러스터의 요구 사항 결정	14
1단계: Memcached 클러스터 시작	14
2단계: 액세스 자격 부여	16
3단계: 클러스터 노드에 연결	16
4단계: 클러스터 삭제(불필요한 요금 부과 방지)	16
추가 정보	17
자습서 및 동영상	18
동영상	18
입문자용 동영상	18
고급 동영상	19
캐싱 전략 및 모범 사례	21
캐싱 전략	21
지연 로딩	21
라이트-스루	23
TTL 추가	24
관련 주제	25
효율적인 로드 밸런싱을 위해 ElastiCache 클라이언트 구성	25
Java를 사용한 일관적 해싱	26
PHP를 사용한 일관적 해싱	26
.NET을 사용한 일관적 해싱	26
ElastiCache for Memcached 구현 관리	27
엔진 버전 및 업그레이드	27
지원되는 Memcached 버전	27
엔진 버전 업그레이드	29
리전 및 가용 영역 선택	30
지원되는 리전 및 엔드포인트	31
노드 찾기	33

유지 관리 관리 중	33
노드 관리	34
노드에 연결	34
예약 노드	36
지원되는 노드 유형	36
노드 교체	42
ElastiCache 클러스터 관리	43
지원되는 Memcached 버전	43
기타 ElastiCache 클러스터 작업	43
클러스터 생성	44
클러스터 세부 정보 보기	72
클러스터 수정	76
클러스터 재부팅	77
클러스터에 노드 추가	79
클러스터에서 노드 제거	83
대기 중인 노드 추가 또는 삭제 작업 취소	87
클러스터 삭제하기	88
클러스터에 액세스	90
클러스터의 플랫폼 확인	90
클러스터에 액세스 권한 부여	92
글로벌 데이터 스토어를 사용한 AWS 리전 간 복제	97
개요	97
사전 조건 및 제한 사항	98
글로벌 데이터 스토리지 사용(콘솔)	99
글로벌 데이터 저장소(CLI) 사용	107
연결 엔드포인트 찾기	108
엔드포인트 찾기(콘솔)	109
엔드포인트 찾기(AWS CLI)	110
엔드포인트 찾기(ElastiCache API)	112
ElastiCache for Memcached 클러스터 조정	113
수평으로 Memcached 조정	113
수직으로 Memcached 조정	114
파라미터 그룹을 사용해 엔진 파라미터 구성	115
파라미터 관리	115
파라미터 그룹 티어	116
파라미터 그룹 생성	116
이름별로 파라미터 그룹 목록 조회	119
파라미터 그룹의 값 목록 조회	123
파라미터 그룹 수정	124
파라미터 그룹 삭제	126
Memcached 특정 파라미터	127
사용 내역, 이벤트 및 비용 모니터링	139
모니터링 사용	139
호스트 수준 지표	139
Memcached 지표	140
어떤 메트릭을 모니터링해야 합니까?	142
메트릭 통계 및 기간 선택	143
Monitoring CloudWatch Cluster and Node Metrics	143
이벤트 모니터링	145
ElastiCache Amazon SNS 알림 관리	145
ElastiCache 이벤트 보기	148
이벤트 알림 및 Amazon SNS	150
태그를 사용한 비용 모니터링	153
콘솔을 사용한 태그 관리	154
AWS CLI를 사용하여 태그 관리	155
ElastiCache API를 사용하여 태그 관리	158
예약 노드를 통해 비용 관리	159

표준 예약 노드 캐시 유형	160
이전 예약 노드 캐시 제공	160
예약 노드 상품에 대한 정보 가져오기	161
예약 노드 구입	164
예약 노드에 대한 정보 가져오기	166
보안	169
데이터 보호	169
인터넷워크 트래픽 개인 정보	170
Amazon VPC 및 ElastiCache 보안	170
서브넷 및 서브넷 그룹	182
보안 그룹: EC2-Classic	188
ID 및 액세스 관리	194
인증	195
액세스 제어	196
액세스 관리 개요	196
자격 증명 기반 정책(IAM 정책) 사용	199
서비스 연결 역할 사용	203
ElastiCache API 권한 참조	209
규정 준수 확인	212
복원성	212
장애 완화	213
인프라 보안	214
셀프 서비스 업데이트	215
서비스 업데이트 관리	215
셀프 서비스 업데이트 적용	215
셀프 서비스 업데이트 중지	220
참조	223
ElastiCache API 사용	223
Query API 사용	223
사용 가능한 라이브러리	225
애플리케이션 문제 해결	226
AWS CloudTrail을 사용하여 Amazon ElastiCache API 호출 로깅	226
ElastiCache용 AWS CLI 설정	230
사전 요구사항	230
명령행 도구 얻기	231
도구 설정	231
도구에 대한 자격 증명 제공	232
환경 변수	233
오류 메시지	233
알림	234
일반 ElastiCache 알림	234
ElastiCache for Memcached 알림	234
설명서 기록	236
AWS Glossary	241

Amazon ElastiCache for Memcached란 무엇입니까?

Amazon ElastiCache for Memcached 사용 설명서에 오신 것을 환영합니다. Amazon ElastiCache는 클라우드에서 분산된 인 메모리 데이터 스토어 또는 캐시 환경을 손쉽게 설정, 관리 및 확장할 수 있는 웹 서비스입니다. 확장 가능하고 비용 효율적인 고성능 캐싱 솔루션을 제공합니다. 또한 분산된 캐시 환경의 배포 및 관리와 관련된 복잡성을 해소할 수 있습니다.

Note

Amazon ElastiCache는 Redis 엔진 및 Memcached 엔진 모두와 함께 작동합니다. 관심 있는 엔진에 대해 설명한 가이드를 사용하십시오. 사용하고 싶은 엔진을 결정하기 어렵다면 이 가이드의 [Memcached와 Redis 비교 \(p. 10\)](#) 단원을 참조하십시오.

Memcached를 사용하는 기존 애플리케이션은 거의 수정하지 않고 ElastiCache를 사용할 수 있습니다. 애플리케이션에서는 사용자가 배포한 ElastiCache 노드의 호스트 이름과 포트 번호에 관한 정보만 필요합니다. Memcached용 ElastiCache Auto Discovery 기능을 사용하면 애플리케이션에서 캐시 클러스터의 모든 노드를 식별하고 이에 연결할 수 있습니다. 이는 사용 가능한 호스트 이름 및 포트 번호의 목록을 유지 관리할 필요가 없음을 뜻합니다. 이러한 방식을 통해 애플리케이션이 클러스터의 노드 멤버십에 대한 변경 사항으로부터 효과적으로 격리됩니다.

ElastiCache for Memcached에는 중요한 프로덕션 배포에 대한 안정성을 향상하는 여러 기능이 있습니다.

- 캐시 노드 실패 자동 감지 및 복구.
- 자동 검색을 위해 클러스터 내 노드 자동 검색 기능 활성화. 따라서 노드를 추가 또는 제거할 때 애플리케이션을 변경할 필요가 없습니다.
- 노드 및 클러스터의 유연한 가용 영역 배치.
- Amazon EC2, Amazon CloudWatch, AWS CloudTrail 및 Amazon SNS 같은 다른 AWS 제품과 통합하여 고성능의 안전한 관리형 인 메모리 캐싱 솔루션 제공

주제

- [일반적인 ElastiCache 사용 사례 및 ElastiCache 활용 방법 \(p. 1\)](#)
- [ElastiCache for Memcached 리소스 \(p. 3\)](#)
- [ElastiCache for Memcached 구성 요소 및 기능 \(p. 4\)](#)
- [구현 관리용 도구 \(p. 8\)](#)

일반적인 ElastiCache 사용 사례 및 ElastiCache 활용 방법

최신 뉴스 또는 제품 카탈로그를 게재할 때나 이벤트 티켓을 판매할 때 가장 중요한 것은 속도입니다. 웹 사이트와 비즈니스의 성공 여부는 콘텐츠를 제공하는 속도에 상당한 영향을 받습니다.

뉴욕 타임즈의 "[For Impatient Web Users, an Eye Blink Is Just Too Long to Wait\(참을성 없는 웹 사용자에게는 눈 깜박하는 시간조차 너무 길게 느껴져\)](#)"라는 기사에 따르면 사용자는 경쟁 사이트 간의 250밀리초(4/4초) 차이를 인지합니다. 사용자는 결국 속도가 느린 사이트를 떠나 빠른 사이트로 이동합니다. [How Webpage Load Time Is Related to Visitor Loss](#)에 따르면 Amazon에서 실시한 테스트에서 로드 시간이 100밀리초(10/10초) 증가할 때마다 매출이 1% 감소하는 결과가 나왔습니다.

다른 사용자가 데이터를 원할 경우 캐시된 데이터를 훨씬 더 빠르게 제공할 수 있습니다. 웹 페이지든 비즈니스 결정을 주도하는 보고서용이든 이는 동일하게 적용되는 사실입니다. 귀사에서 웹 페이지를 캐시하지 않고 지연 시간을 최소화하여 제공할 여유가 있습니까?

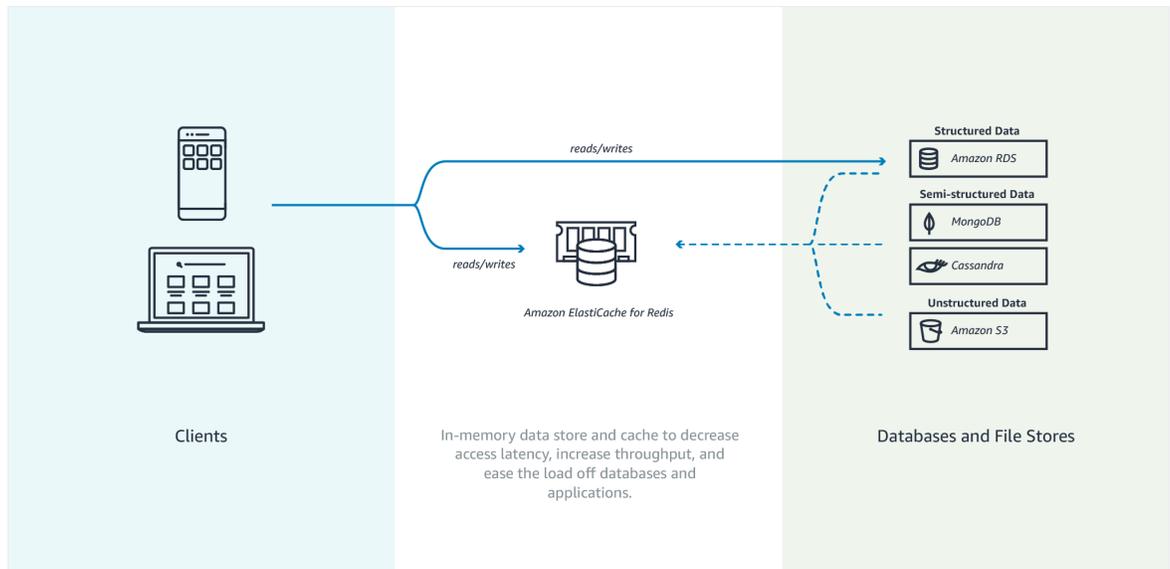
가장 많이 요청되는 항목을 캐시하려 할 것이라는 사실은 어쩌면 자명한 것일 수 있습니다. 요청 빈도가 낮은 항목을 캐시하려고 하지 않는 이유는 무엇입니까? 가장 최적화된 데이터베이스 쿼리나 원격 API 호출이라 할지라도 인 메모리 캐시에서 플랫폼 키를 검색하는 것보다 현저히 느려집니다. 현저히 느려지면 고객이 다른 곳으로 떠나는 경향이 있습니다.

다음 예제에서는 ElastiCache를 사용하여 애플리케이션의 전반적인 성능을 개선할 수 있는 몇 가지 방법을 보여줍니다.

인 메모리 데이터 스토어

인 메모리 키-값 스토어의 주된 목적은 지연 시간이 1밀리초 미만인 매우 빠른 속도와 저렴한 비용으로 데이터 복사본에 액세스할 수 있게 하는 것입니다. 대부분의 데이터 스토어에는 자주 액세스하지만 거의 업데이트하지 않는 데이터 영역이 있습니다. 또한 데이터베이스를 쿼리하면 키-값 페어 캐시에서 키를 찾는 것에 비해 확실히 속도가 느리고 비용이 많이 듭니다. 일부 데이터베이스 쿼리는 특히 수행하는 데 있어 많은 비용이 듭니다. 예로는 여러 표에 걸친 조인이나 복잡한 계산이 포함된 쿼리를 들 수 있습니다. 이러한 쿼리 결과를 캐시하여 쿼리 가격을 한 번만 지불합니다. 그런 다음 쿼리를 다시 실행할 필요 없이 여러 번 데이터를 빠르게 검색할 수 있습니다.

다음 이미지에서는 ElastiCache 캐싱을 보여줍니다.



어떻게 캐시해야 합니까?

캐시할 데이터를 결정할 때 다음과 같은 요인을 고려합니다.

속도 및 비용 – 캐시가 아닌 데이터베이스에서 데이터를 얻는 것이 항상 더 느리고 비용도 많이 듭니다. 다른 데이터베이스 쿼리에 비해 원래 느리고 비용이 많이 드는 데이터베이스 쿼리도 있습니다. 예를 들어, 여러 테이블에서 조인을 수행하는 쿼리는 간단한 싱글 테이블 쿼리에 비해 훨씬 더 느리고 비용이 많이 소요됩니다. 속도가 느리고 비용이 많이 드는 쿼리를 통해서만 얻을 수 있는 데이터라면 캐시가 필요합니다. 비교적 빠르고 간단한 쿼리로 얻을 수 있는 데이터라도 다른 요인에 따라 캐싱이 필요할 수 있습니다.

데이터 및 액세스 패턴 – 캐시할 항목을 결정할 때는 데이터 자체와 액세스 패턴을 이해해야 합니다. 예를 들어, 빠르게 변하거나 거의 액세스하지 않는 데이터는 캐시할 필요가 없습니다. 캐시를 통해 실질적인 이점을 누리려면 비교적 정적이고 자주 액세스하는 데이터여야 합니다. 소셜 미디어 사이트의 개인 프로필을 예로 들 수 있습니다. 반대로, 캐시해도 속도나 비용이 나아지지 않는다면 데이터를 캐시할 필요가 없습니다. 예를

들어 검색 결과를 반환하는 웹 페이지의 쿼리와 결과는 대부분 고유하기 때문에 그러한 웹 페이지를 캐시하는 것은 의미가 없습니다.

기한 경과 - 정의하자면 캐시된 데이터는 기한이 경과한 데이터입니다. 경우에 따라 데이터 기한이 지나지 않았더라도 언제나 기한이 지난 데이터로 간주하고 취급해야 합니다. 데이터를 캐시할지 여부를 결정할 때는 애플리케이션의 데이터 기한 경과 허용 범위를 확인해야 합니다.

애플리케이션에서 기한이 지난 데이터를 허용할 수 있는 상황도 있고 그렇지 않은 상황도 있습니다. 예를 들어 사이트에서 공개적으로 거래된 주가를 제공한다고 가정합니다. 고객이 가격이 n분 지연될 수 있다는 고지 사항과 함께 어느 정도의 기한이 경과할 수 있음을 받아들일 수 있습니다. 하지만 주식을 매각하거나 매입하는 증권인에게 주식의 가격을 제공할 때는 실시간 데이터가 필요합니다.

다음에 해당하는 경우 데이터 캐시를 고려하십시오.

- 캐시 검색에 비해 느린 속도와 높은 비용으로 데이터를 얻습니다.
- 사용자가 자주 데이터에 액세스합니다.
- 데이터가 비교적 동일하게 유지되거나 빠르게 변경되어도 기한 경과가 큰 문제가 되지 않습니다.

자세한 내용은 다음 자료를 참조하십시오.

- ElastiCache for Memcached 사용 설명서의 [캐싱 전략](#)

ElastiCache 고객 추천사

Airbnb, PBS, Esri 등의 기업이 어떻게 Amazon ElastiCache을 활용해 고객 환경을 개선하여 비즈니스 성장을 도모하고 있는지에 관해서는 [추천사 Amazon ElastiCache](#)를 참조하십시오.

다른 ElastiCache 고객 사용 사례는 [ElastiCache 동영상 \(p. 18\)](#)에서 보실 수 있습니다.

ElastiCache for Memcached 리소스

먼저 다음의 단원을 읽은 다음, 필요에 따라 참조하는 것이 좋습니다.

- 서비스 하이라이트 및 요금 - [제품 세부 정보 페이지](#)는 일반적인 ElastiCache 제품 개요, 서비스 하이라이트 및 요금 정보를 제공합니다.
- ElastiCache 비디오 - [ElastiCache 동영상 \(p. 18\)](#) 단원에는 Memcached용 Amazon ElastiCache를 소개하고, 일반 사용 사례를 다루며, ElastiCache for Memcached를 사용하여 지연 시간을 줄이고 애플리케이션의 처리량을 향상하는 방법을 설명하는 비디오가 포함되어 있습니다.
- 시작하기 - [Amazon ElastiCache for Memcached 시작하기 \(p. 13\)](#) 단원에는 캐시 클러스터 생성, 캐시 클러스터에 대한 액세스 인증, 캐시 노드 연결 및 캐시 클러스터 삭제 과정을 안내하는 예가 포함되어 있습니다.
- 규모에 따른 성능 - [Amazon ElastiCache의 규모에 따른 성능](#) 백서에서는 규모에 따라 애플리케이션이 원활하게 수행할 수 있는 캐싱 전략에 대해 설명합니다.

이전 단원을 완료한 후에는 다음 단원을 읽어 보십시오.

- [Memcached 노드 크기 선택 \(p. 68\)](#)

캐시하려는 모든 데이터를 수용할 만큼의 충분한 노드를 원하며, 동시에 필요한 것보다 더 많은 캐시의 비용을 지불하고 싶지 않은 경우 이 항목을 사용하여 최적의 노드 크기를 선택할 수 있습니다.

- [캐싱 전략 및 모범 사례 \(p. 21\)](#)

클러스터의 효율성에 영향을 줄 수 있는 문제를 확인하고 해결하십시오.

AWS Command Line Interface (AWS CLI)를 사용할 경우 시작하는 데 도움이 되도록 다음 문서를 사용할 수 있습니다.

- [AWS Command Line Interface 설명서](#)

이 섹션에서는 AWS CLI 다운로드, 시스템에서 작동 중인 AWS CLI 가져오기 및 AWS 자격 증명 제공에 대한 정보를 제공합니다.

- [AWS CLI ElastiCache용 설명서](#)

이 별도의 문서는 구문 및 예제를 포함한 모든 ElastiCache용 AWS CLI 명령을 다룹니다.

ElastiCache API를 널리 사용되는 다양한 프로그래밍 언어로 사용하도록 애플리케이션 프로그램을 작성할 수 있습니다. 다음과 같은 몇 가지 리소스가 있습니다.

- [Amazon Web Services용 도구](#)

Amazon Web Services은 Memcached용 ElastiCache에 대한 지원과 더불어 다양한 SDK(소프트웨어 개발 키트)를 제공합니다. Java, .NET, PHP, Ruby 및 기타 언어를 사용하여 ElastiCache에 대해 코딩할 수 있습니다. 이러한 SDK는 ElastiCache에 대한 요청을 포맷하고 응답을 분석하며 재시도 논리 및 오류 처리를 제공함으로써 애플리케이션 개발을 상당히 간소화할 수 있습니다.

- [ElastiCache API 사용 \(p. 223\)](#)

AWS SDK를 사용하지 않으려는 경우 Query API를 직접 사용하면서 ElastiCache와 상호 작용할 수 있습니다. 이 단원에서 요청 생성과 인증 및 응답 처리에 대한 정보와 문제 해결 팁을 찾을 수 있습니다.

- [Amazon ElastiCache API Reference](#)

이 별도의 문서는 구문 및 예제를 포함한 모든 ElastiCache API 작업을 다룹니다.

ElastiCache for Memcached 구성 요소 및 기능

다음은 Memcached용 Amazon ElastiCache 배포의 주요 구성 요소를 개괄적으로 설명한 것입니다.

주제

- [ElastiCache 노드 \(p. 4\)](#)
- [ElastiCache for Memcached 클러스터 \(p. 5\)](#)
- [AWS 리전 및 가용 영역 \(p. 6\)](#)
- [ElastiCache for Memcached 엔드포인트 \(p. 6\)](#)
- [ElastiCache 파라미터 그룹 \(p. 7\)](#)
- [ElastiCache 보안 \(p. 7\)](#)
- [ElastiCache 보안 그룹 \(p. 7\)](#)
- [ElastiCache 서브넷 그룹 \(p. 7\)](#)
- [ElastiCache for Memcached 이벤트 \(p. 8\)](#)

ElastiCache 노드

노드는 ElastiCache 배포의 가장 작은 구성 요소입니다. 노드는 다른 노드와 독립적으로 존재하거나 다른 노드와 일부 관련되어 존재할 수 있습니다.

노드는 안전한 네트워크에 연결된 RAM의 크기가 고정된 청크입니다. 각 노드는 Memcached의 인스턴스를 실행합니다. 필요한 경우 클러스터의 노드를 다른 인스턴스 유형으로 스케일 업하거나 스케일 다운할 수 있습니다. 자세한 내용은 [ElastiCache for Memcached 클러스터 조정 \(p. 113\)](#) 단원을 참조하십시오.

클러스터 내 모든 노드는 인스턴스 유형이 동일하며, 동일한 캐시 엔진을 실행합니다. 각 캐시 노드에는 고유한 DNS(Domain Name Service) 이름 및 포트가 있습니다. 여러 유형의 캐시 노드가 지원되며 연결된 메모리 양이 각각 다릅니다. 지원되는 노드 인스턴스 유형의 목록은 [지원되는 노드 유형 \(p. 36\)](#) 단원을 참조하십시오.

사용한 노드에 대해서만 비용을 지불하는 "사용한 만큼 지불"을 기준으로 노드를 구입할 수 있습니다. 또는 대폭 인하된 시간당 요금으로 예약 노드를 구입할 수 있습니다. 사용률이 높은 경우, 예약 노드를 구입하면 비용을 절약할 수 있습니다. 클러스터가 항상 사용 중에 있고 사용량 폭증을 처리하기 위해 가끔 노드를 추가하는 경우를 생각해 봅시다. 이 경우에는 지속적인 클러스터 실행을 위해 여러 예약 노드를 구입하고, 가끔 노드를 추가해야 하는 경우를 위해 종량 요금제 노드를 구입할 수 있습니다. 예약 노드에 대한 자세한 내용은 [ElastiCache 예약 노드 \(p. 36\)](#) 단원을 참조하십시오.

Memcached 엔진은 자동 검색을 지원합니다. Auto Discovery는 클라이언트 프로그램이 캐시 클러스터의 모든 노드를 자동으로 식별하고 이 모든 노드에 대한 연결을 시작하고 유지 관리할 수 있는 기능입니다. Auto Discovery 기능 덕분에 애플리케이션에서는 개별 노드에 수동으로 연결할 필요가 없습니다. 그 대신에 애플리케이션은 구성 엔드포인트에 연결됩니다. 구성 엔드포인트 DNS 항목에는 캐시 노드 엔드포인트 각각에 대한 CNAME 항목이 포함되어 있습니다. 따라서 구성 엔드포인트에 연결되는 즉시 애플리케이션은 클러스터의 모든 노드에 관한 정보를 얻게 되어 모든 노드에 연결할 수 있습니다. 애플리케이션에서 각각의 캐시 노드 엔드포인트를 하드 코딩할 필요가 없습니다. Auto Discovery에 대한 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오.

노드에 대한 자세한 내용은 [노드 관리 \(p. 34\)](#) 단원을 참조하십시오.

ElastiCache for Memcached 클러스터

Memcached 클러스터는 [ElastiCache 노드 \(p. 4\)](#) 하나 이상의 논리적 그룹입니다. 데이터는 Memcached 클러스터의 노드로 분할됩니다.

여러 ElastiCache 작업은 클러스터에서 다음 사항을 대상으로 합니다.

- 클러스터 생성
- 클러스터 수정
- 클러스터 삭제
- 클러스터의 요소 보기
- 비용 할당 태그를 클러스터에 추가 및 클러스터에서 삭제

자세한 내용은 다음 관련 항목을 참조하십시오.

- [ElastiCache 클러스터 관리 \(p. 43\)](#) 및 [노드 관리 \(p. 34\)](#)

클러스터, 노드 및 관련 작업에 대한 정보입니다.

- [AWS 서비스 제한: Amazon ElastiCache](#)

노드 또는 클러스터의 최대 수와 같은 ElastiCache 제한에 대한 정보입니다.

이러한 제한을 초과해야 할 경우 [Amazon ElastiCache 캐시 노드 요청 양식](#)을 사용하여 요청하십시오.

- [장애 완화 \(p. 213\)](#)

클러스터의 내결함성 향상에 관한 정보입니다.

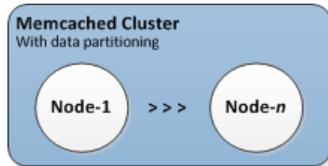
일반적인 클러스터 구성

Memcached는 AWS 리전마다 고객당 최대 100개의 노드를 지원하는데, 각 클러스터에는 -20개의 노드가 있습니다. 사용자는 데이터를 Memcached 클러스터의 노드에 두루 분할합니다.

Memcached 엔진을 실행할 경우 클러스터는 1~20개의 노드로 구성될 수 있습니다. 사용자는 데이터베이스를 노드에 두루 분할합니다. 애플리케이션은 각 노드의 엔드포인트를 읽고 씁니다. 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오.

향상된 내결함성을 위해 클러스터의 AWS 리전 내에 있는 다양한 AZ(가용 영역)에서 Memcached 노드를 찾습니다. 이렇게 하면 한 AZ에서 발생한 오류가 전체 클러스터 및 애플리케이션에 미치는 영향을 최대한 줄일 수 있습니다. 자세한 내용은 [장애 완화 \(p. 213\)](#) 단원을 참조하십시오.

Memcached 클러스터 변경 시 요구 사항에 따라 노드를 추가 또는 제거하여 확장 또는 축소할 수 있습니다. 이로써 데이터는 새 노드 수에 두루 재분할됩니다. 데이터를 분할할 때 일관적 해싱을 사용하는 것이 좋습니다. 일관적 해싱에 대한 자세한 내용은 [효율적인 로드 밸런싱을 위해 ElastiCache 클라이언트 구성 \(p. 25\)](#) 단원을 참조하십시오. 다음 다이어그램에는 단일 노드 및 다중 노드 Memcached 클러스터의 예시가 있습니다.

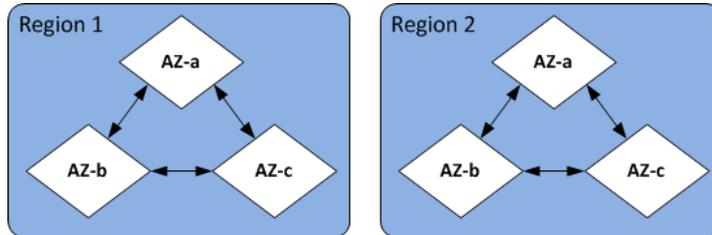


AWS 리전 및 가용 영역

Memcached용 Amazon ElastiCache는 전 세계 여러 AWS 리전에서 사용할 수 있습니다. 따라서 비즈니스 요구 사항에 적합한 위치에서 ElastiCache 클러스터를 시작할 수 있습니다. 예를 들어 고객과 가장 가깝거나 특정 법적 요구 사항을 준수하는 AWS 리전에서 시작할 수 있습니다.

기본적으로 AWS SDK, AWS CLI, ElastiCache API 및 ElastiCache 콘솔은 미국 서부(오레곤) 리전을 참조합니다. ElastiCache가 새 리전에서 사용할 수 있게 되면서 이러한 리전의 새 엔드포인트는 AWS 요청, HTTP SDK, AWS CLI 및 ElastiCache 콘솔에서도 사용할 수 있습니다.

각 AWS 리전은 다른 AWS 리전에서 완전히 격리되도록 설계되었습니다. 각 리전 내에는 가용 영역이 여러 개 있습니다. 서로 다른 가용 영역에서 노드를 시작하면 가능한 최고 수준의 내결함성을 갖출 수 있습니다. AWS 리전 및 가용 영역에 대한 자세한 내용은 [리전 및 가용 영역 선택 \(p. 30\)](#) 단원을 참조하십시오.



ElastiCache에서 지원되는 AWS 리전 및 해당 엔드포인트에 대한 자세한 내용은 [지원되는 리전 및 엔드포인트 \(p. 31\)](#) 단원을 참조하십시오.

ElastiCache for Memcached 엔드포인트

엔드포인트는 애플리케이션이 ElastiCache 노드 또는 클러스터에 연결하는 데 사용하는 고유한 주소입니다.

Memcached 클러스터의 각 노드에는 고유한 엔드포인트가 있습니다. 클러스터에도 구성 엔드포인트라는 엔드포인트가 있습니다. Auto Discovery를 사용하기로 설정하고 구성 엔드포인트를 연결한 경우에는 클러스터에서 노드를 추가하거나 제거한 후에도 애플리케이션이 자동으로 각 노드 엔드포인트를 파악합니다. 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오.

자세한 내용은 [연결 엔드포인트 찾기 \(p. 108\)](#) 단원을 참조하십시오.

ElastiCache 파라미터 그룹

캐시 파라미터 그룹은 지원되는 엔진 소프트웨어에 대한 런타임 설정을 관리하는 간단한 방법입니다. 메모리 사용량, 제거 정책, 항목 크기 등을 제어하는 데 여러 가지 파라미터가 사용됩니다. ElastiCache 파라미터 그룹이란 클러스터에 적용할 수 있는 엔진별 파라미터의 이름이 지정된 모음을 말합니다. 이를 통해 해당 클러스터에 있는 모든 노드가 정확히 동일한 방법으로 구성되게 할 수 있습니다.

지원되는 파라미터의 목록, 해당 기본값 및 수정할 수 있는 사항은 [DescribeEngineDefaultParameters](#) 단원 ([describe-engine-default-parameters](#))을 참조하십시오.

ElastiCache 파라미터 그룹에 대한 자세한 내용은 [파라미터 그룹을 사용해 엔진 파라미터 구성 \(p. 115\)](#) 단원을 참조하십시오.

ElastiCache 보안

향상된 보안을 위해 ElastiCache 노드 액세스는 화이트리스트로 지정된 Amazon EC2 인스턴스에 실행되는 애플리케이션으로 제한됩니다. 서브넷 그룹 또는 보안 그룹을 사용하여 클러스터에 액세스할 수 있는 Amazon EC2 인스턴스를 제어할 수 있습니다.

기본적으로 모든 새 ElastiCache 클러스터는 Amazon Virtual Private Cloud(Amazon VPC) 환경에서 시작됩니다. 특정 서브넷에서 실행 중인 Amazon EC2 인스턴스로부터 클러스터 액세스를 부여하려면 서브넷 그룹을 사용하면 됩니다. Amazon VPC 외부에서 클러스터를 실행하도록 선택할 경우 보안 그룹을 생성하여 특정 Amazon EC2 보안 그룹 내에서 실행되는 Amazon EC2 인스턴스를 승인할 수 있습니다.

ElastiCache 보안 그룹

Note

ElastiCache 보안 그룹은 Amazon Virtual Private Cloud(Amazon VPC) 환경에서 실행되지 않는 클러스터에만 적용됩니다. ElastiCache 노드를 Amazon VPC에서 실행할 경우 ElastiCache 보안 그룹과 다른 Amazon VPC 보안 그룹으로 캐시 클러스터에 대한 액세스를 제어합니다. Amazon VPC에서 ElastiCache 사용에 대한 자세한 내용은 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

ElastiCache에서는 보안 그룹을 통해 클러스터에 대한 액세스를 제어할 수 있습니다. 보안 그룹은 클러스터에 대한 네트워크 액세스를 제어하는 방화벽 역할을 합니다. 기본적으로 클러스터로 액세스하는 네트워크는 해제되어 있습니다. 애플리케이션이 클러스터에 액세스하도록 하려면 특정 Amazon EC2 보안 그룹의 호스트에서 액세스를 명시적으로 활성화해야 합니다. 수신 규칙이 구성되면 동일한 규칙이 해당 보안 그룹과 관련된 모든 클러스터에 적용됩니다.

클러스터에 대한 네트워크 액세스를 허용하려면 먼저 보안 그룹을 생성합니다. 그런 다음 [AuthorizeCacheSecurityGroupIngress](#) API 작업 또는 [authorize-cache-security-group-ingress](#) AWS CLI 명령을 사용하여 원하는 Amazon EC2 보안 그룹을 승인합니다. 이렇게 하면 결과적으로 허용되는 Amazon EC2 인스턴스가 지정됩니다. 보안 그룹은 생성 시에, 또는 ElastiCache 관리 콘솔, [ModifyCacheCluster](#) 또는 ([modify-cache-cluster](#))을 사용하여 클러스터에 보안 그룹을 연결할 수 있습니다. ElastiCache용 AWS CLI 명령.

Important

현재 IP 범위 기반 액세스 제어는 클러스터에 사용할 수 없습니다. 클러스터에 액세스하는 모든 클라이언트는 Amazon EC2 네트워크 내에 있어야 하며, 이전에 설명한 대로 보안 그룹을 통해 승인을 받아야 합니다.

보안 그룹에 대한 자세한 내용은 [보안 그룹: EC2-Classical \(p. 188\)](#) 단원을 참조하십시오.

ElastiCache 서브넷 그룹

서브넷 그룹은 Amazon Virtual Private Cloud(Amazon VPC) 환경에서 실행 중인 클러스터에 대해 지정할 수 있는 서브넷(일반적으로 프라이빗 서브넷) 모음입니다.

Amazon VPC에서 클러스터를 생성할 경우 캐시 서브넷 그룹을 지정해야 합니다. ElastiCache는 해당 캐시 서브넷 그룹을 사용하여 캐시 노드에 연결된 서브넷 내의 서브넷 및 IP 주소를 선택합니다.

Amazon VPC 환경에서 캐시 서브넷 그룹 사용에 대한 자세한 내용은 [Amazon VPC 및 ElastiCache 보안](#) (p. 170), [2단계: 액세스 자격 부여](#) (p. 16) 및 [서브넷 및 서브넷 그룹](#) (p. 182) 단원을 참조하십시오.

ElastiCache for Memcached 이벤트

캐시 클러스터에서 중요 이벤트가 발생하면 ElastiCache는 특정 Amazon SNS 주제에 알림을 전송합니다. 중요 이벤트로는 노드 추가 실패, 노드 추가 성공, 보안 그룹 수정 등을 들 수 있습니다. 주요 이벤트를 모니터링하면 클러스터의 현재 상태를 파악할 수 있으며, 이벤트에 따라 교정 작업을 수행할 수도 있습니다.

ElastiCache 이벤트에 대한 자세한 내용은 [ElastiCache 이벤트 모니터링](#) (p. 145) 단원을 참조하십시오.

구현 관리용 도구

ElastiCache 클러스터에 대한 Amazon EC2 인스턴스 액세스를 허용하면 ElastiCache 클러스터를 관리할 수 있는 네 가지 수단(AWS Management 콘솔, ElastiCache용 AWS CLI, ElastiCache용 AWS SDK 및 ElastiCache API)이 제공됩니다.

AWS Management 콘솔 사용

가장 간편한 Amazon ElastiCache for Memcached 관리 방법은 AWS Management 콘솔을 사용하는 것입니다. 콘솔을 사용하면 캐시 클러스터를 생성하고 캐시 노드를 추가 및 제거하며 코드를 작성하지 않고도 다른 관리 작업을 수행할 수 있습니다. 콘솔은 캐시 엔진 활동, 메모리 및 CPU 사용률과 기타 측정치를 보여주는 CloudWatch의 캐시 노드 성능 그래프도 제공합니다. 자세한 내용은 이 사용 설명서의 특정 항목을 참조하십시오.

AWS CLI 사용

ElastiCache에서 AWS Command Line Interface(AWS CLI)를 사용해도 됩니다. AWS CLI를 사용하면 캐시 클러스터 시작 또는 중지와 같은 한 번에 한 작업을 쉽게 수행할 수 있습니다. 선택한 스크립팅 언어에서 ElastiCache용 AWS CLI 명령어를 호출하여 반복 작업을 자동화할 수도 있습니다. AWS CLI에 대한 자세한 내용은 사용 설명서 및 [AWS CLI Command Reference](#)를 참조하십시오.

AWS SDK 사용

애플리케이션에서 ElastiCache에 액세스하려면 AWS SDK(소프트웨어 개발 키트) 중 하나를 사용할 수 있습니다. SDK는 ElastiCache API 호출을 래핑하고 ElastiCache API의 하위 수준 세부 정보로부터 애플리케이션을 보호합니다. 사용자는 자격 증명을 제공하고 SDK 라이브러리가 인증 및 요청 서명을 담당합니다. AWS SDK에 대한 자세한 내용은 [Amazon Web Services용 도구](#)를 참조하십시오.

ElastiCache API 사용

또한 ElastiCache 웹 서비스 API에 대해 직접 애플리케이션 코드를 작성할 수 있습니다. API를 사용할 때 HTTP 요청을 구성하고 인증하고 ElastiCache의 결과를 분석하고 오류를 처리하기 위해 필요한 코드를 작성해야 합니다. API에 대한 자세한 내용은 [ElastiCache API 사용](#) (p. 223)을 참조하십시오.

다음 사항도 참조하세요.

Amazon ElastiCache for Memcached 배포 관리에 대한 자세한 내용은 다음을 참조하십시오.

- [ElastiCache for Memcached 구현 관리](#) (p. 27)

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
다음 사항도 참조하세요.

- [인터넷워크 트래픽 개인 정보 \(p. 170\)](#)
- [사용 내역, 이벤트 및 비용 모니터링 \(p. 139\)](#)

Memcached와 Redis 비교

Amazon ElastiCache는 Memcached 및 Redis 캐시 엔진을 지원합니다. 각 엔진에는 몇 가지 장점이 있습니다. 이 항목의 정보를 활용하면 요구 사항에 가장 잘 맞는 엔진과 버전을 선택하는 데 도움이 됩니다.

Important

캐시 클러스터 또는 복제 그룹을 생성한 후에는 최신 엔진 버전으로 업그레이드할 수 있지만, 이전 엔진 버전으로 다운그레이드할 수는 없습니다. 이전 엔진 버전을 사용하려면 기존 캐시 클러스터 또는 복제 그룹을 삭제하고 이전 엔진 버전을 통해 다시 생성해야 합니다.

표면적으로는 엔진이 유사하게 보입니다. 각 엔진은 인 메모리 키-값 저장소입니다. 그러나 실제로 상당한 차이점이 있습니다.

다음과 같은 경우 Memcached를 선택합니다.

- 가능한 가장 단순한 모델이 필요한 경우
- 여러 코어 또는 스레드가 있는 큰 노드를 실행해야 하는 경우
- 시스템의 요구 사항이 증가하고 감소함에 따라 노드를 추가 및 제거하는 확장 및 축소 기능이 필요한 경우
- 데이터베이스와 같은 객체를 캐시에 저장해야 하는 경우

다음과 같은 경우 Redis용 ElastiCache 버전이 있는 Redis를 선택합니다.

- Redis용 ElastiCache 버전 5.0.0(개선됨)

생산자가 실시간으로 새 항목을 추가하고 소비자가 차단 또는 비 차단 방식으로 메시지를 사용할 수 있도록 지원하는 로그 데이터 구조인 [Redis 스트림](#)을 사용할 수 있습니다.

자세한 내용은 [Redis 버전 5.0.0\(개선됨\)](#)을 참조하십시오.

- Redis용 ElastiCache 버전 4.0.10(개선됨)

암호화 및 Redis(클러스터 모드 활성화됨) 클러스터에서 샤드의 동적으로 추가 또는 제거를 지원합니다.

자세한 내용은 [Redis 버전 4.0.10\(개선됨\)](#)을 참조하십시오.

- Redis용 ElastiCache 버전 3.2.10(개선됨)

Redis(클러스터 모드 활성화됨) 클러스터에서 샤드를 동적으로 추가 또는 제거하는 기능을 지원합니다.

Important

Redis용 ElastiCache 3.2.10에서는 현재 암호화가 지원되지 않습니다.

자세한 내용은 다음 자료를 참조하십시오.

- [Redis 버전 3.2.10\(개선됨\)](#)
- Redis에 대한 온라인 리샤딩 모범 사례에 대한 자세한 내용은 다음을 참조하십시오.
 - [모범 사례: 온라인 리샤딩](#)
 - [Redis\(클러스터 모드 활성화됨\)를 위한 온라인 리샤딩 및 샤드 재분배](#)
- Redis 클러스터 조정에 대한 자세한 내용은 [조정 단원](#)을 참조하십시오.
- Redis용 ElastiCache 버전 3.2.6(개선됨)

다음 기능 이외에 이전 Redis 버전의 기능이 필요하면 Redis용 ElastiCache 3.2.6을 선택하십시오.

- 전송 중 데이터 암호화. 자세한 내용은 [Redis용 Amazon ElastiCache 전송 중인 데이터 암호화](#) 단원을 참조하십시오.
- 미사용 데이터 암호화. 자세한 내용은 [Redis용 Amazon ElastiCache 미사용 데이터 암호화](#) 단원을 참조하십시오.
- HIPAA 자격 획득 인증 자세한 내용은 [Redis용 Amazon ElastiCache에 대한 HIPAA 자격 획득](#) 단원을 참조하십시오.
- Redis용 ElastiCache (클러스터 모드 활성화됨) 버전 3.2.4

다음 기능 이외에 Redis 2.8.x의 기능이 필요하면 Redis 3.2.4(클러스터 모드)를 선택합니다.

- 2 ~ 90개의 노드 그룹으로 데이터를 분할해야 하는 경우(클러스터 모드에만 해당)
- 지역 검색 인덱싱이 필요한 경우(클러스터 모드 또는 비클러스터 모드)
- 여러 데이터베이스를 지원할 필요가 없는 경우

Important

Redis(클러스터 모드 활성화됨)에는 다음과 같은 제한 사항이 있습니다.

- 더 큰 노드 유형으로 확장할 수 없음
- 노드 그룹의 복제본 수를 변경할 수 없음(분할)

- Redis용 ElastiCache(비 클러스터 모드) 2.8x 및 3.2.4(개선됨)

다음과 같은 경우 Redis 2.8.x 또는 Redis 3.2.4(비클러스터 모드)를 선택합니다.

- 문자열, 해시, 목록, 세트, 정렬된 세트 및 비트맵과 같은 복잡한 데이터 유형이 필요한 경우
- 인 메모리 데이터 세트를 정렬하거나 순위를 지정해야 하는 경우
- 키 저장소의 지속성을 원할 경우
- 읽기 집약적 애플리케이션을 위해 기본 항목에서 하나 이상의 읽기 전용 복제본으로 데이터를 복제해야 하는 경우
- 기본 노드가 실패할 때 자동 장애 조치가 필요한 경우
- 서버에 대한 이벤트를 클라이언트에 알리기 위해 게시 및 구독(게시/구독) 기능이 필요합니다.
- 백업 및 복원 기능이 필요한 경우
- 여러 데이터베이스를 지원해야 하는 경우

Memcached, Redis(클러스터 모드 비활성화됨) 및 Redis(클러스터 모드 활성화됨)의 비교 요약

	Memcached	Redis(클러스터 모드 비활성화됨)	Redis(클러스터 모드 활성화됨)
엔진 버전	1.4.x	2.8.x 이상	3.2.x 이상
데이터 유형	간단함	2.8.x - 복합 * 복합	3.2.x 이상 - 복합
데이터 파티셔닝	예	아니요	예
클러스터 수정 가능	예	예	3.2.10이상 - 제한
온라인 리샤딩	아니요	아니요	3.2.10 이상
암호화	아니요	3.2.6, 4.0.10 및 이상	3.2.6, 4.0.10 및 이상
규정 준수 인증			
규정 준수 인증			
FedRAMP	아니요	2015-02-02 2.6, 4.0.10 및 이상	3.2.6, 4.0.10 및 이상

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서

	Memcached	Redis(클러스터 모드 비활성화됨)	Redis(클러스터 모드 활성화됨)
HIPAA	아니요	3.2.6, 4.0.10 및 이상	3.2.6, 4.0.10 및 이상
PCI DSS	아니요	3.2.6, 4.0.10 및 이상	3.2.6, 4.0.10 및 이상
다중 스레드	예	아니요	아니요
노드 유형 업그레이드	아니요	예	예
엔진 업그레이드	예	예	예
고가용성(복제)	아니요	예	예
자동 장애조치(failover)	아니요	선택	필수
게시/구독 기능	아니요	예	예
정렬된 세트	아니요	예	예
백업 및 복원	아니요	예	예
지역 검색 인덱싱	아니요	2.8.x - 아니요	예
		3.2.x 이상 - 예	
참고:			
문자열, 객체(예: 데이터베이스)			
* 문자열, 세트, 정렬된 세트, 목록, 해시, 비트맵, HyperLogLog			
문자열, 세트, 정렬된 세트, 목록, 해시, 비트맵, hyperloglog, 지역 검색 인덱스			

클러스터에 대한 엔진을 선택한 후 해당 엔진의 최신 버전을 사용하는 것이 좋습니다. 자세한 내용은 [지원 ElastiCache for Memcached 버전](#) 또는 [지원 Redis용 ElastiCache 버전](#)을 참조하십시오.

Amazon ElastiCache for Memcached 시작하기

이 단원의 여러 주제에서는 ElastiCache 콘솔을 사용하여 Memcached 클러스터를 생성하는 것부터 이 클러스터에 액세스 권한을 부여하고 연결하며 마지막으로 이 클러스터를 삭제하는 데까지 일련의 과정을 안내해 드립니다.

주제

- [설정 \(p. 13\)](#)
- [Memcached 클러스터 배포 \(p. 14\)](#)
- [추가 정보 \(p. 17\)](#)

설정

다음 주제에서는 ElastiCache 사용을 시작하기 위해 수행해야 하는 1회성 작업에 대해 설명합니다.

주제

- [AWS 계정 생성 \(p. 13\)](#)
- [권한 설정\(신규 ElastiCache 사용자에게만 해당\) \(p. 13\)](#)

AWS 계정 생성

Amazon ElastiCache를 사용하려면 활성 AWS 계정과 ElastiCache 및 기타 AWS 리소스에 액세스할 수 있는 권한이 있어야 합니다.

아직 AWS 계정이 없다면 사용할 계정을 하나 생성해야 합니다. AWS 계정은 무료입니다. AWS 서비스에 가입한 것 자체로는 요금이 청구되지 않고 사용한 AWS 서비스에 대해서만 청구됩니다.

AWS 계정을 생성하려면

1. <https://portal.aws.amazon.com/billing/signup>을 엽니다.
2. 온라인 지시 사항을 따릅니다.

등록 절차 중 전화를 받고 전화 키패드를 사용하여 확인 코드를 입력하는 과정이 있습니다.

권한 설정(신규 ElastiCache 사용자에게만 해당)

Amazon ElastiCache에서는 서비스 연결 역할을 생성 및 사용해 리소스를 프로비저닝하고 사용자를 대신해 다른 AWS 리소스 및 서비스에 액세스합니다. ElastiCache에서 사용자를 위해 서비스 연결 역할을 생성하도록 하려면 AmazonElastiCacheFullAccess라는 AWS 관리형 정책을 사용하십시오. 이 역할은 서비스가 사용자를 대신해 서비스 연결 역할을 생성하는 데 필요한 권한으로 사전에 프로비저닝되어 있습니다.

사용자는 기본 정책을 사용하는 대신 사용자 지정 관리형 정책을 사용하는 쪽을 선택할 수 있습니다. 이 경우 iam:createServiceLinkedRole을 호출할 권한이 있거나 ElastiCache 서비스 연결 역할을 생성했어야 합니다.

자세한 내용은 다음을 참조하십시오.

- [새 정책 생성\(IAM\)](#)

- [Amazon ElastiCache에 대한 AWS 관리형\(미리 정의된\) 정책 \(p. 201\)](#)
- [Amazon ElastiCache에 서비스 연결 역할 사용 \(p. 203\)](#)

Memcached 클러스터 배포

이제 AWS 계정 및 권한을 설정했으므로 클러스터를 배포하여 ElastiCache for Memcached를 사용해볼 수 있습니다. 다음 단원에서는 그 방법을 안내합니다.

주제

- [클러스터의 요구 사항 결정 \(p. 14\)](#)
- [1단계: Memcached 클러스터 시작 \(p. 14\)](#)
- [2단계: 액세스 자격 부여 \(p. 16\)](#)
- [3단계: 클러스터 노드에 연결 \(p. 16\)](#)
- [4단계: 클러스터 삭제\(불필요한 요금 부과 방지\) \(p. 16\)](#)

클러스터의 요구 사항 결정

Memcached 클러스터를 생성하기 전에 항상 클러스터 요구 사항을 결정하여 클러스터를 생성할 때 비즈니스 요구를 충족하게 함으로써 다시 생성할 필요가 없도록 해야 합니다. 이 연습에서는 대체로 클러스터에 기본값을 적용할 것이므로 요구 사항 확인은 생략하겠습니다. 자세한 내용은 [요구 사항 결정 \(p. 44\)](#) 단원을 참조하십시오.

1단계: Memcached 클러스터 시작

시작하려는 클러스터가 활성화되고 샌드박스에서 실행되지 않습니다. 삭제하기 전까지 인스턴스에 대해 표준 ElastiCache 사용 요금이 청구됩니다. 여기에 설명된 연습을 한 번에 끝내고 연습을 마칠 때 클러스터를 삭제하면 총 청구 비용이 가장 적게 듭니다(일반적으로 1달러 미만). ElastiCache 사용 요금에 대한 자세한 내용은 <https://aws.amazon.com/elasticache/>를 참조하십시오.

Important

클러스터는 Amazon VPC에서 시작됩니다. 클러스터 생성을 시작하기 전에 서브넷 그룹을 생성해야 합니다. 자세한 내용은 [서브넷 그룹 생성 \(p. 183\)](#) 단원을 참조하십시오.

Memcached 클러스터용 ElastiCache를 생성하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. [Get Started Now]를 선택합니다.

사용할 수 있는 클러스터가 이미 있는 경우 [Launch Cluster]를 선택합니다.

3. 상단 오른쪽 모서리의 목록에서 이 클러스터를 시작할 AWS 리전을 선택합니다.
4. [Cluster engine]에 대해 [Memcached]를 선택합니다.
5. 다음과 같이 [Memcached settings] 섹션을 완료합니다.
 - a. [Name]에 클러스터 이름을 입력합니다.

클러스터 명명 제약 조건은 다음과 같습니다.

- 1-40자의 영숫자 또는 하이픈으로 구성되어야 합니다.
- 문자로 시작해야 합니다.
- 하이픈 2개가 연속될 수 없습니다.

• 끝에 하이픈이 올 수 없습니다.

- b. [Engine version compatibility] 목록에서 이 클러스터를 실행할 Memcached 엔진 버전을 선택합니다. 이전 버전을 실행할 특별한 이유가 없는 한 최신 버전을 선택하는 것이 좋습니다.
- c. [Port]에서 기본 포트 11211을 허용합니다. 다른 포트를 사용하려면 포트 번호를 입력합니다.
- d. [Parameter group]에서 이 클러스터에서 사용할 파라미터 그룹을 선택하거나 "새로 생성"을 선택하여 새 파라미터 그룹을 생성하여 해당 클러스터에서 사용합니다. 이 연습에서는 기본 파라미터 그룹을 사용합니다.

자세한 내용은 [파라미터 그룹 생성 \(p. 116\)](#) 단원을 참조하십시오.

- e. [Node type]에 대해 이 클러스터에 사용할 노드 유형을 선택합니다. 이 연습에서는 기본 노드 유형을 수락하거나 다른 노드 유형을 선택할 수 있습니다.

자세한 내용은 [Memcached 노드 크기 선택 \(p. 68\)](#) 단원을 참조하십시오.

다른 노드 유형을 선택하려면

- i. 기본 노드 유형의 오른쪽에 있는 아래쪽 화살표를 선택합니다.
- ii. 이 클러스터에 있는 노드에 대해 원하는 인스턴스 패밀리를 선택합니다. 연습일 뿐이므로 비용을 절약하기 위해 t2를 선택합니다.
- iii. 사용 가능한 노드 유형에서 이 클러스터에 대해 원하는 노드 유형의 왼쪽에 있는 확인란을 선택합니다. 연습일 뿐이므로 비용을 절약하기 위해 cache.t2.small을 선택합니다.
- iv. Save를 선택합니다.

- f. 노드 수 목록에서 이 클러스터에 프로비저닝되기를 원하는 노드 수(파티션)를 선택합니다.

6. [Advanced Memcached settings]를 선택하고 다음과 같이 섹션을 완료합니다.

- a. [Subnet group] 목록에서 클러스터에 적용할 서브넷을 선택합니다. 이 연습에서는 기본 서브넷 그룹을 적용합니다.

자세한 내용은 [서브넷 및 서브넷 그룹 \(p. 182\)](#) 단원을 참조하십시오.

- b. [Availability zone(s)]에 대해 다음 두 가지 옵션이 있습니다.

- 기본 설정 없음 – ElastiCache에서 사용자를 위해 각 노드의 가용 영역을 선택합니다.
- Specify availability zones(가용 영역 설정) – 각 노드의 가용 영역을 지정합니다.

이 연습에서는 [Specify availability zones]를 선택한 다음 각 노드에 대해 가용 영역을 선택합니다.

자세한 내용은 [리전 및 가용 영역 선택 \(p. 30\)](#) 단원을 참조하십시오.

- c. [Security groups] 목록에서 이 클러스터에서 사용할 보안 그룹을 선택합니다. 이 연습에서는 기본 보안 그룹을 적용합니다.

자세한 내용은 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

- d. 유지 관리 기간은 ElastiCache가 클러스터에서 시스템 유지 관리를 예약하는 시간이며 일반적으로 매주 한 시간입니다. 여기에는 두 가지 옵션이 있습니다.

- 기본 설정 없음—ElastiCache에서는 클러스터의 유지 관리 기간에 대해 요일과 시간을 선택합니다.
- Specify maintenance window(유지 관리 기간 지정)—클러스터의 유지 관리 기간에 대해 요일, 시작 시간 및 기간을 선택합니다.

클러스터가 생성된 후에는 이를 수정하여 다른 유지 관리 기간을 지정할 수 있습니다. 자세한 내용은 [유지 관리 관리 중 \(p. 33\)](#) 단원을 참조하십시오.

- e. 알림은 알림 비활성화 상태로 둡니다.

7. [Create]를 선택하여 클러스터를 시작하거나 [Cancel]을 선택해 작업을 취소합니다.

2단계: 액세스 자격 부여

이 단원에서는 Amazon EC2 인스턴스를 시작하고 연결하는 것에 익숙하다고 가정합니다. 자세한 내용은 [Amazon EC2 Getting Started Guide](#) 섹션을 참조하십시오.

모든 ElastiCache 클러스터는 Amazon EC2 인스턴스에서 액세스하도록 설계되었습니다. 가장 일반적인 시나리오는 동일한 Amazon Virtual Private Cloud(Amazon VPC)의 Amazon EC2 인스턴스에서 ElastiCache 클러스터에 액세스하는 것입니다. 이 항목에서는 시나리오를 다룹니다. 다른 Amazon VPC, 다른 리전 또는 기업 네트워크에서 ElastiCache 클러스터를 액세스에 대한 정보는 다음을 참조하십시오.

- [Amazon VPC에 있는 ElastiCache 클러스터에 액세스하기 위한 액세스 패턴 \(p. 174\)](#)
- [AWS 외부에서 ElastiCache 리소스 액세스 \(p. 93\)](#)

기본적으로 클러스터에 대한 네트워크 액세스는 이를 시작하는 데 사용한 사용자 계정으로 제한됩니다. EC2 인스턴스에서 클러스터에 연결하려면 EC2 인스턴스가 클러스터에 액세스하도록 권한을 부여해야 합니다. 필요한 단계는 클러스터를 EC2-VPC 또는 EC2-Classic으로 시작했는지에 따라 다릅니다.

클러스터에 액세스할 수 있는 권한을 부여하는 절차는 [클러스터에 액세스 \(p. 90\)](#) 단원을 참조하십시오.

3단계: 클러스터 노드에 연결

계속하기 전에 먼저 [2단계: 액세스 자격 부여 \(p. 16\)](#)을 완료해야 합니다.

이 단원에서는 Amazon EC2 인스턴스를 생성했고 이 인스턴스에 연결할 수 있다고 가정합니다. 작업 방법에 대한 지침은 [Amazon EC2 Getting Started Guide](#) 단원을 참조하십시오.

Amazon EC2 인스턴스는 권한을 부여한 경우에만 클러스터 노드에 연결할 수 있습니다. 자세한 내용은 [2단계: 액세스 자격 부여 \(p. 16\)](#) 단원을 참조하십시오.

3.1단계: 노드 엔드포인트 찾기

클러스터가 사용 가능한 상태이고 사용자가 이 클러스터에 액세스할 수 있는 권한을 부여받았다면([2단계: 액세스 자격 부여 \(p. 16\)](#)) Amazon EC2 인스턴스에 로그인하여 클러스터에 연결할 수 있습니다. 이를 수행하려면 먼저 엔드포인트를 결정해야 합니다.

엔드포인트를 찾으려면 다음 목록에서 관련 주제를 참조하십시오. 필요한 엔드포인트를 찾은 경우 3.2단계에서 사용할 수 있도록 클립보드에 복사합니다.

- [연결 엔드포인트 찾기 \(p. 108\)](#)
- [클러스터의 엔드포인트 찾기 \(콘솔\) \(p. 109\)](#)—클러스터의 구성 엔드포인트가 필요합니다.
- [엔드포인트 찾기 \(AWS CLI\) \(p. 110\)](#)
- [엔드포인트 찾기 \(ElastiCache API\) \(p. 112\)](#)

3.2단계: Memcached 클러스터에 연결

클러스터가 사용 가능한 상태이고 액세스 권한을 부여받으면([2단계: 액세스 자격 부여 \(p. 16\)](#)) Amazon EC2 인스턴스에 로그인하여 클러스터에 연결할 수 있습니다.

Memcached 클러스터 연결에 관한 지침은 [노드에 연결 \(p. 34\)](#) 단원을 참조하십시오.

4단계: 클러스터 삭제(불필요한 요금 부과 방지)

Important

활발하게 사용하지 않는 클러스터는 삭제하는 것이 거의 언제나 바람직합니다. 클러스터의 상태가 [deleted]가 되기 전까지 요금이 계속 부과됩니다.

계속하기 전에 최소한 1단계: [Memcached 클러스터 시작 \(p. 14\)](#) 단계를 완료해야 합니다.

Memcached 클러스터를 삭제하려면

1. AWS Management 콘솔에 로그인한 후 <https://console.aws.amazon.com/elasticache/>에서 Amazon ElastiCache 콘솔을 엽니다.
2. Memcached를 실행 중인 모든 클러스터의 목록을 보려면 탐색 창에서 Memcached를 선택합니다.
3. 삭제할 클러스터를 선택하려면 클러스터 목록에서 클러스터의 이름을 선택합니다.

Tip

ElastiCache 콘솔에서는 클러스터를 한 번에 하나씩만 삭제할 수 있습니다. 여러 클러스터를 선택하면 삭제 작업이 비활성화됩니다. 여러 클러스터를 삭제하려면 클러스터마다 이 절차를 반복하십시오.

4. [Actions]에 대해 [Delete]를 선택합니다.
5. 클러스터 삭제 확인 화면에서 삭제를 선택하여 클러스터를 삭제하거나 취소를 선택하여 클러스터를 유지합니다.

삭제를 선택하면 클러스터의 상태가 삭제 중으로 바뀝니다.

클러스터가 클러스터 목록에서 제거되는 즉시 요금 부과가 중단됩니다.

이로써 첫 번째 ElastiCache for Memcached 클러스터를 시작하여 이에 대한 액세스 권한을 부여하고, 연결, 확인, 삭제하는 일련의 과정을 마쳤습니다.

추가 정보

이제 시작하기 연습을 끝마쳤으므로 다음 단원을 참조하여 ElastiCache 및 사용 가능한 도구에 대한 자세한 내용을 살펴볼 수 있습니다.

- [AWS 시작](#)
- [Amazon Web Services용 도구](#)
- [AWS 명령줄 인터페이스](#)
- [Amazon ElastiCache API 참조](#)

시작하기 연습을 마친 후에는 다음 단원을 읽어 ElastiCache 관리에 대해 좀 더 자세히 알아볼 수 있습니다.

- [Memcached 노드 크기 선택 \(p. 68\)](#)

캐시하려는 모든 데이터를 수용할 만큼의 충분한 캐시를 원하며, 동시에 필요한 것보다 더 많은 캐시의 비용을 지불하고 싶지 않은 경우 이 단원을 읽어보면 최적의 노드 크기를 선택하는 데 도움이 됩니다.

- [캐싱 전략 및 모범 사례 \(p. 21\)](#)

클러스터의 효율성에 부정적인 영향을 미칠 수 있는 문제를 확인하고 해결하십시오.

ElastiCache 자습서 및 동영상

다음은 Amazon ElastiCache 사용자가 관심을 기울이는 작업을 해결하는 자습서입니다.

- [ElastiCache 동영상 \(p. 18\)](#)
- [자습서: Amazon VPC에서 Amazon ElastiCache에 액세스하도록 Lambda 함수 구성](#)

ElastiCache 동영상

다음으로, 기본 및 고급 Amazon ElastiCache 개념을 학습할 수 있는 동영상을 찾을 수 있습니다. AWS 교육에 대한 정보는 [AWS 교육 및 자격증](#)을 참조하십시오.

주제

- [입문자용 동영상 \(p. 18\)](#)
- [고급 동영상 \(p. 19\)](#)

입문자용 동영상

다음 비디오는 Amazon ElastiCache에 대해 소개합니다.

주제

- [DAT204—AWS NoSQL 서비스에 확장 가능한 애플리케이션 구축\(re:Invent 2015\) \(p. 18\)](#)
- [DAT207—Amazon ElastiCache를 통해 애플리케이션 성능 가속화\(AWS re:Invent 2013\) \(p. 18\)](#)

DAT204—AWS NoSQL 서비스에 확장 가능한 애플리케이션 구축 (re:Invent 2015)

이 세션에서는 NoSQL 데이터베이스에 대한 이점을 설명하고, AWS에서 제공한 기본 NoSQL 서비스 (Amazon DynamoDB 및 Amazon ElastiCache)에 대해 알아보겠습니다. 그런 다음, 두 선도적인 고객인 Expedia와 Mapbox로부터 사용 사례와 아키텍처 문제점 및 설계 패턴과 모범 사례를 비롯한 AWS NoSQL 서비스를 사용하여 이를 해결한 방법에 대해 듣습니다. 이 세션을 통해 NoSQL과 NoSQL의 강력한 기능을 잘 이해하여 데이터베이스 문제점을 확실하게 처리할 준비를 갖추게 됩니다.

[DAT204—AWS NoSQL 서비스에 확장 가능한 애플리케이션 구축\(re:Invent 2015\)](#)

DAT207—Amazon ElastiCache를 통해 애플리케이션 성능 가속화 (AWS re:Invent 2013)

이 동영상에서는 애플리케이션 성능의 속도를 높이기 위해 Amazon ElastiCache를 사용하여 인 메모리 캐시를 쉽게 배포하는 방법에 대해 알아보십시오. Amazon ElastiCache를 사용하여 애플리케이션 지연 시간을 개선하고 데이터베이스 서버에 대한 부담을 줄이는 방법을 알아보십시오. 또한 애플리케이션이 성장함에 따라 쉽게 관리 및 조정할 수 있는 캐싱 레이어를 구축하는 방법도 알아보십시오. 이 세션 중 캐싱을 활성화하여 혜택을 받을 수 있는 다양한 시나리오 및 사용 사례를 살펴보고, Amazon ElastiCache에서 제공한 기능을 설명합니다.

[DAT207 - Amazon ElastiCache를 통해 애플리케이션 성능 가속화\(re:Invent 2013\)](#)

고급 동영상

다음 비디오는 고급 Amazon ElastiCache 주제를 다룹니다.

주제

- [DAT305—Amazon ElastiCache 심층 분석\(re:Invent 2017\)](#) (p. 19)
- [DAT306—Amazon ElastiCache 심층 분석\(re:Invent 2016\)](#) (p. 19)
- [DAT407—Amazon ElastiCache 심층 분석\(re:Invent 2015\)](#) (p. 19)
- [SDD402—Amazon ElastiCache 심층 분석\(re:Invent 2014\)](#) (p. 19)
- [DAT307—Amazon ElastiCache 아키텍처 및 설계 패턴에 대한 심층 분석\(re:Invent 2013\)](#) (p. 20)

DAT305—Amazon ElastiCache 심층 분석(re:Invent 2017)

Amazon ElastiCache의 설계 및 아키텍처에 대한 뒷이야기를 살펴봅니다. Memcached 및 Redis 솔루션으로 일반적인 설계 패턴을 살펴보고, 고객이 인 메모리 작업에 이런 패턴을 사용하여 지연 시간을 단축하고 애플리케이션 처리량을 개선한 방법을 알아봅니다. 이 비디오 중에는 ElastiCache 모범 사례, 설계 패턴 및 안티 패턴을 검토합니다.

비디오는 다음의 내용을 소개합니다.

- Redis용 ElastiCache 온라인 리샤딩
- ElastiCache 보안 및 암호화
- Redis용 ElastiCache 버전 3.2.10

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

DAT306—Amazon ElastiCache 심층 분석(re:Invent 2016)

Amazon ElastiCache의 설계 및 아키텍처에 대한 뒷이야기를 살펴봅니다. Memcached 및 Redis 솔루션으로 일반적인 설계 패턴을 살펴보고, 고객이 인 메모리 작업에 이런 패턴을 사용하여 지연 시간을 단축하고 애플리케이션 처리량을 개선한 방법을 알아봅니다. 이 세션 중에는 ElastiCache 모범 사례, 설계 패턴 및 안티 패턴을 검토합니다.

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

DAT407—Amazon ElastiCache 심층 분석(re:Invent 2015)

Amazon ElastiCache의 설계 및 아키텍처에 대한 뒷이야기를 살펴봅니다. Memcached 및 Redis 솔루션의 일반적인 설계 패턴을 살펴보고, 고객이 인 메모리 작업에 이런 패턴을 사용하여 애플리케이션에 대한 개선된 지연 시간 및 처리량을 실현한 방법을 알아봅니다. 이 세션 중에는 Amazon ElastiCache와 관련된 모범 사례, 설계 패턴, 안티 패턴을 검토합니다.

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

SDD402—Amazon ElastiCache 심층 분석(re:Invent 2014)

이 동영상에서는 일반적인 캐싱 사용 사례, Memcached 및 Redis 엔진, 요구 사항에 적합한 엔진을 결정하는 데 도움이 되는 패턴, 일관적 해싱 및 기타 빠르고 확장 가능한 애플리케이션을 구축하는 수단을 살펴봅니다. Adobe의 최고 책임 과학자인 Frank Wiebe는 Adobe에서 Amazon ElastiCache를 사용하여 고객 경험을 향상하고 비즈니스를 확장하는 방법을 자세히 설명합니다.

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

DAT307—Amazon ElastiCache 아키텍처 및 설계 패턴에 대한 심층 분석(re:Invent 2013)

이 동영상에서는 캐싱, 캐싱 전략, 확장, 모니터링을 살펴봅니다. 또한 Memcached 및 Redis 엔진을 비교합니다. 이 세션 중에는 Amazon ElastiCache와 관련된 모범 사례 및 설계 패턴도 검토합니다.

[DAT307 - Amazon ElastiCache 아키텍처 및 설계 패턴에 대한 심층 분석\(AWS re:Invent 2013\)](#).

캐싱 전략 및 모범 사례

다음은 Amazon ElastiCache에 대한 권장 모범 사례입니다. 다음 모범 사례를 준수하면 클러스터의 성능과 안정성이 향상됩니다.

주제

- [캐싱 전략 \(p. 21\)](#)
- [효율적인 로드 밸런싱을 위해 ElastiCache 클라이언트 구성 \(p. 25\)](#)

캐싱 전략

다음 항목에서는 캐시를 채우고 유지 관리하기 위한 전략을 확인할 수 있습니다.

캐시를 채우고 유지 관리하기 위해 구현하려는 전략은 캐싱되는 데이터의 유형과 해당 데이터에 대한 액세스 패턴에 따라 달라집니다. 예를 들어, 게임 사이트와 새 이야기 추세의 상위 10개 리더보드에 대해 동일한 전략을 사용하려고 하지 않을 수 있습니다. 이 단원의 나머지 부분에서는 일반적인 캐시 유지 관리 전략, 이에 대한 장점 및 단점에 대해 살펴봅니다.

주제

- [지연 로딩 \(p. 21\)](#)
- [라이트-스루 \(p. 23\)](#)
- [TTL 추가 \(p. 24\)](#)
- [관련 주제 \(p. 25\)](#)

지연 로딩

이름에서 알 수 있듯이 지연 로딩은 필요할 때에만 데이터를 캐시에 로드하는 캐싱 전략입니다. 다음 설명과 같이 작동합니다.

Amazon ElastiCache는 액세스하는 애플리케이션과 데이터 스토어(데이터베이스) 사이에 위치하는 인 메모리 키/값 저장소입니다. 애플리케이션에서 데이터를 요청할 때마다 ElastiCache 캐시에 먼저 요청합니다. 데이터가 캐시에 있으며 최신 상태인 경우 ElastiCache가 데이터를 애플리케이션에 반환합니다. 데이터가 캐시에 없거나 만료된 경우 애플리케이션에서 데이터 스토어에 데이터를 요청합니다. 데이터 스토어에서 데이터를 애플리케이션에 반환합니다. 다음으로 애플리케이션은 스토어에서 수신한 데이터를 캐시에 씁니다. 이렇게 하면 다음에 요청이 있을 때 데이터를 더 빨리 검색할 수 있습니다.

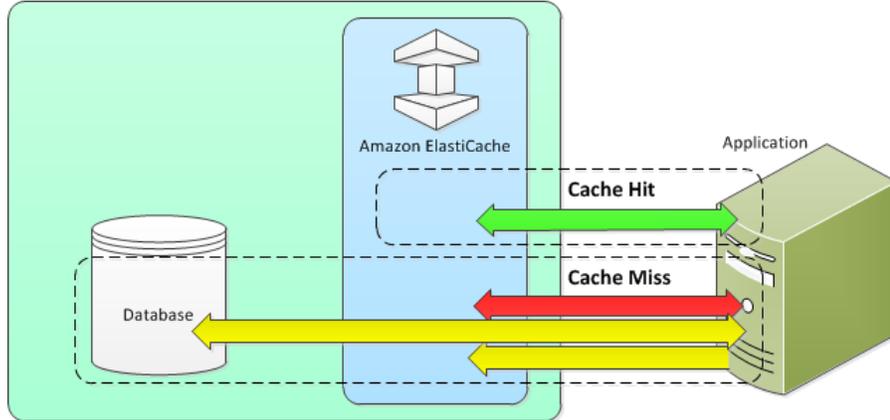
캐시 적중률은 데이터가 캐시에 있고 만료되지 않은 경우에 발생합니다.

1. 애플리케이션은 캐시에서 데이터를 요청합니다.
2. 캐시는 애플리케이션으로 데이터를 반환합니다.

캐시 누락은 데이터가 캐시에 없거나 만료된 경우에 발생합니다.

1. 애플리케이션은 캐시에서 데이터를 요청합니다.
2. 캐시에 데이터가 요청되지 않으므로 `null`을 반환합니다.
3. 애플리케이션은 데이터베이스에 데이터를 요청하고 수신합니다.
4. 애플리케이션은 새 데이터로 캐시를 업데이트합니다.

다음 다이어그램은 이러한 프로세스를 모두 보여줍니다.



지연 로딩의 장점 및 단점

지연 로딩의 장점은 다음과 같습니다.

- 요청된 데이터만 캐싱됩니다.

대부분의 데이터가 요청되지 않으므로 지연 로딩은 요청되지 않은 데이터가 있는 캐시를 채우지 않습니다.

- 노드 장애가 애플리케이션에 치명적인 영향을 주지 않습니다.

노드 장애가 발생하여 새로운 빈 노드로 대체될 경우 애플리케이션에서는 지연 시간 증가를 통해 계속 작동합니다. 새 노드에 대한 요청이 발생하면, 각 캐시가 누락될 때마다 데이터베이스 쿼리가 생성됩니다. 동시에 데이터 복사본이 캐시에 추가되어 이후의 요청이 캐시에서 검색됩니다.

지연 로딩의 단점은 다음과 같습니다.

- 캐시 누락 패널티가 있습니다. 각 캐시 누락은 세 개의 이동으로 나타납니다.

- 캐시에서 데이터에 대한 초기 요청
- 데이터에 대한 데이터베이스의 쿼리
- 캐시에 데이터 작성

이러한 누락으로 인해 애플리케이션으로 데이터를 가져오는 것이 눈에 띄게 지연될 수 있습니다.

- 기한 경과된 데이터

캐시 누락 시에만 데이터를 캐시에 쓰면, 캐시의 데이터가 기한이 경과할 수 있습니다. 이 결과는 데이터베이스에서 데이터가 변경될 때 캐시에 대한 업데이트가 없기 때문에 발생합니다. 이 문제를 해결하려면 [라이트-스루](#) (p. 23) 및 [TTL 추가](#) (p. 24) 전략을 사용할 수 있습니다.

지연 로딩 의사(pseudo) 예제

다음은 지연 로딩 로직의 의사(pseudo) 코드 예제입니다.

```
// *****  
// 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
```

이 예제의 경우, 데이터를 갖는 애플리케이션 코드는 다음과 같습니다.

```
customer_record = get_customer(12345)
```

라이트-스루

라이트-스루 전략은 데이터베이스에 데이터를 작성할 때마다 데이터를 추가하거나 캐시의 데이터를 업데이트합니다.

라이트-스루의 장점 및 단점

라이트-스루의 장점은 다음과 같습니다.

- 캐시의 데이터가 기한 경과되지 않습니다.

캐시의 데이터는 데이터베이스에 쓰일 때마다 업데이트되므로 항상 최신 상태입니다.

- 쓰기 패널티 vs. 읽기 패널티

모든 쓰기에는 다음 2개의 이동이 수반됩니다.

1. 캐시에 쓰기
2. 데이터베이스에 쓰기

이로 인해 프로세스에 지연 시간이 추가됩니다. 즉, 최종 사용자는 일반적으로 데이터를 검색할 때보다 데이터를 업데이트할 때 지연 시간에 더 관대합니다. 업데이트는 더 많이 작동하므로 오래 걸릴 수 있다는 고유한 생각이 있습니다.

라이트-스루의 단점은 다음과 같습니다.

- 누락된 데이터

노드 장애 또는 확장으로 인해 새 노드를 스핀업하면 데이터가 누락됩니다. 이 데이터는 데이터베이스에 추가되거나 업데이트될 때까지 계속 누락됩니다. 라이트-스루로 [지연 로딩 \(p. 21\)](#)을 구현하여 이 문제를 최소화할 수 있습니다.

- 캐시 이탈

대부분의 데이터는 절대 읽히지 않으며 이는 리소스 낭비입니다. [TTL\(Time to Live\) 값을 추가 \(p. 24\)](#)하면 낭비되는 공간을 최소화할 수 있습니다.

라이트-스루 의사(pseudo) 예제

다음은 라이트-스루 로직의 의사(pseudo) 코드 예제입니다.

```
// *****  
// 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
```

이 예제의 경우, 데이터를 갖는 애플리케이션 코드는 다음과 같습니다.

```
save_customer(12345, {"address": "123 Main"})
```

TTL 추가

지연 로딩은 기한 경과 데이터에 대해 허용되지만 빈 노드로 인해 실패하지 않습니다. 라이트-스루는 데이터를 항상 최신 상태로 유지하지만, 빈 노드로 인해 실패할 수 있으며 불필요한 데이터로 캐시를 채울 수 있습니다. 각 쓰기에 TTL(Time to Live) 값을 추가하면 각 전략의 이점을 얻을 수 있습니다. 동시에 추가 데이터로 캐시를 복잡하게 만들지 않을 수 있습니다.

TTL(Time To Live)은 키가 만료될 때까지의 시간(초)을 지정하는 정수 값입니다. Redis는 이 값에 대해 초 또는 밀리초를 지정할 수 있습니다. 애플리케이션에서 만료된 키를 읽으려고 하면 키가 없는 것으로 처리됩니다. 데이터베이스가 키에 대해 쿼리되고 캐시가 업데이트됩니다. 이는 값이 기한 경과가 아님을 보장하지 않습니다. 그러나 데이터가 너무 기간 경과되지 않도록 방지하며, 경우에 따라 캐시의 값이 데이터베이스에서 새로 고침되어야 합니다.

자세한 내용은 또는 [Memcached set 명령을 참조하십시오](#).

TTL 의사(pseudo) 예제

다음 코드는 TTL을 통한 라이트-스루 로직의 의사(pseudo) 코드 예제입니다.

```
// *****  
// 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
```

다음은 TTL을 통한 지연 로딩 로직의 의사(pseudo) 코드 예제입니다.

```
// *****  
// 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
```

이 예제의 경우, 데이터를 갖는 애플리케이션 코드는 다음과 같습니다.

```
save_customer(12345, {"address": "123 Main"})
```

```
customer_record = get_customer(12345)
```

관련 주제

- [인 메모리 데이터 스토어 \(p. 2\)](#)
- [엔진 및 버전 선택](#)
- [ElastiCache for Memcached 클러스터 조정 \(p. 113\)](#)

효율적인 로드 밸런싱을 위해 ElastiCache 클라이언트 구성

Note

이 단원의 내용은 다중 노드 Memcached 클러스터에 적용됩니다.

여러 ElastiCache Memcached 노드를 효과적으로 사용하려면 캐시 키를 노드에 분산시킬 수 있어야 합니다. 노드가 n 개인 클러스터의 로드 밸런싱을 유지하려면 객체 키의 해시를 계산하고 $n - \text{hash}(\text{key}) \bmod n$ 으로 결과를 모드하는 것이 간편합니다. 결과 값 (0 ~ $n-1$)은 객체가 배치되는 노드의 수입니다.

노드 수(n)가 일정하면 이 방법이 간편하고 효과적입니다. 하지만 클러스터에서 노드를 추가하거나 제거할 때마다 이동할 키의 수는 $(n - 1) / n$ 입니다. 여기서 n 은 새로운 노드 수입니다. 따라서 이 방법을 사용하면 많은 키가 이동하며 특히 노드 수가 커질수록 다량의 초기 캐시가 누락됩니다. 1개에서 2개로 노드를 조정하면 이동할 키는 $(2-1) / 2(50\%)$ 가 되는 것이 모범 사례입니다. 9개에서 10개로 노드를 조정하면 이동할 키는 $(10-1)/10(90\%)$ 가 됩니다. 트래픽의 스파이크로 인해 확장하는 경우 캐시가 대량으로 누락되는 상황을 방지할 필요가 없습니다. 캐시가 대량으로 누락되면 데이터베이스에 대한 히트가 발생하지만 이미 트래픽의 스파이크로 인해 오버로드되어 있습니다.

이 문제의 해결 방법은 일관적 해싱입니다. 일관적 해싱에는 클러스터에서 노드가 추가되거나 제거될 때마다 이동할 키의 수가 대략 $1/n$ (여기서 n 은 새로운 노드 수)인 알고리즘이 사용됩니다. 1개에서 2개로 노드를 조정하면 이동할 키는 $1/2(50\%)$ 이 되는 것이 최악의 경우입니다. 9개에서 10개로 노드를 조정하면 이동할 키는 $1/10(10\%)$ 가 됩니다.

사용자는 다중 노드 클러스터에 사용되는 해시 알고리즘을 제어합니다. 일관적 해싱을 사용하도록 클라이언트를 구성하는 것이 좋습니다. 일관적 해싱을 구현하는 데 자주 사용되는 언어로 여러 Memcached 클라이언트 라이브러리가 제공됩니다. 사용할 라이브러리의 설명서에서 일관적 해싱 지원 여부와 그 구현 방법을 참조하십시오.

Java, PHP 또는 .NET으로 작업하는 경우 Amazon ElastiCache 클라이언트 라이브러리 중 하나를 사용하는 것이 좋습니다.

Java를 사용한 일관적 해싱

ElastiCache Memcached Java 클라이언트는 일관적 해싱 기능이 내장된 오픈 소스 spymemcached Java 클라이언트를 기반으로 합니다. 일관적 해싱을 구현하는 KetamaConnectionFactory 클래스가 라이브러리에 포함되어 있습니다. 기본적으로 spymemcached에서는 일관적 해싱이 해제됩니다.

자세한 내용은 <http://dustin.sallings.org/java-memcached-client/apidocs/net/spy/memcached/KetamaConnectionFactory.html>에서 KetamaConnectionFactory 설명서를 참조하십시오.

PHP를 사용한 일관적 해싱

ElastiCache Memcached PHP 클라이언트는 기본 제공 Memcached PHP 라이브러리 주변의 래퍼입니다. 기본적으로 Memcached PHP 라이브러리에서는 일관적 해싱이 해제됩니다.

다음 코드를 사용하여 일관적 해싱을 설정하십시오.

```
$m = new Memcached();  
$m->setOption(Memcached::OPT_DISTRIBUTION, Memcached::DISTRIBUTION_CONSISTENT);
```

또한 php.ini 파일에서 memcached.sess_consistent_hash를 설정할 수도 있습니다.

자세한 내용은 <http://php.net/manual/en/memcached.configuration.php>에서 Memcached PHP의 런타임 구성 설명서를 참조하십시오. 특히 memcached.sess_consistent_hash 파라미터에 유의하시기 바랍니다.

.NET을 사용한 일관적 해싱

ElastiCache Memcached .NET 클라이언트는 Enyim Memcached 주변의 래퍼입니다. 기본적으로 Enyim Memcached 클라이언트에서는 일관적 해싱이 설정됩니다.

자세한 내용은 <https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator>의 memcached/locator 설명서를 참조하십시오.

ElastiCache for Memcached 구현 관리

이 단원에서는 ElastiCache 구현의 다양한 구성 요소를 관리하는 방법에 관한 세부 정보를 얻을 수 있습니다. 여기에는 노드나 클러스터를 생성, 업데이트 및 삭제하는 등의 작업이 포함됩니다.

주제

- [엔진 버전 및 업그레이드 \(p. 27\)](#)
- [리전 및 가용 영역 선택 \(p. 30\)](#)
- [유지 관리 관리 중 \(p. 33\)](#)
- [노드 관리 \(p. 34\)](#)
- [ElastiCache 클러스터 관리 \(p. 43\)](#)
- [클러스터에 액세스 \(p. 90\)](#)
- [글로벌 데이터 스토어를 사용한 AWS 리전 간 복제 \(p. 97\)](#)
- [연결 엔드포인트 찾기 \(p. 108\)](#)
- [ElastiCache for Memcached 클러스터 조정 \(p. 113\)](#)
- [파라미터 그룹을 사용해 엔진 파라미터 구성 \(p. 115\)](#)

엔진 버전 및 업그레이드

이 단원에서는 지원되는 Memcached 엔진 버전과 이를 업그레이드하는 방법을 설명합니다.

주제

- [지원되는 ElastiCache for Memcached 버전 \(p. 27\)](#)
- [엔진 버전 업그레이드 \(p. 29\)](#)

지원되는 ElastiCache for Memcached 버전

ElastiCache는 다음 Memcached 버전과 새 버전으로의 업그레이드를 지원합니다. 새 버전으로 업그레이드할 때 충족되지 않을 경우 업그레이드가 실패하는 조건에 주의를 기울이십시오.

Memcached 버전용 ElastiCache

- [Memcached 버전 1.5.16 \(p. 27\)](#)
- [Memcached 버전 1.5.10 \(p. 28\)](#)
- [Memcached 버전 1.4.34 \(p. 28\)](#)
- [Memcached 버전 1.4.33 \(p. 28\)](#)
- [Memcached 버전 1.4.24 \(p. 29\)](#)
- [Memcached 버전 1.4.14 \(p. 29\)](#)
- [Memcached 버전 1.4.5 \(p. 29\)](#)

Memcached 버전 1.5.16

Memcached용 ElastiCache에 Memcached 버전 1.5.16에 대한 지원이 추가되었습니다. 새로운 기능은 없지만 [Memcached 1.5.14](#) 및 [Memcached 1.5.15](#)의 버그 수정 및 누적 업데이트가 포함되어 있습니다.

자세한 내용은 GitHub의 Memcached에서 [Memcached 1.5.16 출시 정보](#)를 참조하십시오.

Memcached 버전 1.5.10

Memcached용 ElastiCache 버전 1.5.10은 다음 Memcached 기능을 지원합니다.

- 자동화된 슬래브 재분배 기능.
- murmur3 알고리즘으로 더 빠른 해시 테이블 조회.
- 세분화된 LRU 알고리즘.
- background-reclaim 메모리에 대한 LRU 크롤러.
- `--enable-seccomp`: 컴파일 시간 옵션.

또한 `no_modern` 및 `inline_ascii_resp` 파라미터를 도입합니다. 자세한 내용은 [Memcached 1.5.10 파라미터 변경 \(p. 128\)](#) 단원을 참조하십시오.

Memcached용 ElastiCache 버전 1.4.34부터 추가된 Memcached 개선 사항에는 다음이 포함됩니다.

- ASCII multiget, CVE-2017-9951 및 `metadumper`에 대한 크롤링 한도와 같은 누적 방식 수정.
- 연결 한도에서 연결을 닫는 방식으로 연결 관리 향상.
- 1MB 이상의 항목 크기에 대한 항목 크기 관리 개선.
- 항목당 메모리 요구 사항을 몇 바이트 줄임으로써 성능 및 메모리 오버헤드 개선.

자세한 내용은 GitHub의 Memcached에서 [Memcached 1.5.10 출시 정보](#)를 참조하십시오.

Memcached 버전 1.4.34

Memcached 버전 1.4.34용 ElastiCache는 버전 1.4.33에 대해 새 기능이 추가되지 않았습니다. 버전 1.4.34는 일반적인 릴리스보다 큰 버그 수정 릴리스입니다.

자세한 내용은 GitHub의 Memcached에서 [Memcached 1.4.34 출시 정보](#)를 참조하십시오.

Memcached 버전 1.4.33

버전 1.4.24부터 추가된 Memcached 개선 사항에는 다음이 포함됩니다.

- 특정 슬래브 클래스, 슬래브 클래스 목록 또는 모든 슬래브 클래스에 대한 모든 메타데이터를 덤프할 수 있습니다. 자세한 내용은 [Memcached 1.4.31 출시 정보](#)를 참조하십시오.
- 1메가바이트 기본값보다 큰 항목에 대한 지원이 개선되었습니다. 자세한 내용은 [Memcached 1.4.29 출시 정보](#)를 참조하십시오.
- 종료하라는 메시지가 표시되기 전에 클라이언트가 유휴 상태로 있을 수 있는 기간을 지정할 수 있습니다.

클러스터를 다시 시작하지 않고 Memcached에 사용할 수 있는 메모리의 양을 동적으로 늘릴 수 있습니다. 자세한 내용은 [Memcached 1.4.27 출시 정보](#)를 참조하십시오.

- 이제 `fetchers`, `mutations` 및 `evictions`의 로깅이 지원됩니다. 자세한 내용은 [Memcached 1.4.26 출시 정보](#)를 참조하십시오.
- 빈 메모리를 전역 풀로 다시 회수하여 새 슬래브 클래스로 재할당할 수 있습니다. 자세한 내용은 [Memcached 1.4.25 출시 정보](#)를 참조하십시오.
- 여러 가지 버그 수정.
- 일부 새 명령 및 파라미터. 목록을 보려면 [Memcached 1.4.33 추가 파라미터 \(p. 129\)](#) 단원을 참조하십시오.

Memcached 버전 1.4.24

버전 1.4.14부터 추가된 Memcached 개선 사항에는 다음이 포함됩니다.

- 백그라운드 프로세스를 통한 LRU(가장 오랫동안 사용되지 않음) 관리.
- 해시 알고리즘으로 jenkins 또는 murmur3의 옵션이 추가되었습니다.
- 일부 새 명령 및 파라미터. 목록을 보려면 [Memcached 1.4.24 추가 파라미터 \(p. 130\)](#) 단원을 참조하십시오.
- 여러 가지 버그 수정.

Memcached 버전 1.4.14

버전 1.4.5부터 추가된 Memcached 개선 사항에는 다음이 포함됩니다.

- 슬래브 재분배 기능이 개선되었습니다.
- 성능 및 확장성 개선.
- 기존 항목을 가져오지 않고 해당 항목의 만료 시간을 업데이트하기 위해 터치 명령이 도입되었습니다.
- Auto Discovery — 클라이언트 프로그램이 클러스터의 모든 캐시 노드를 자동으로 확인하고 이러한 모든 노드에 대한 연결을 시작하고 유지 관리할 수 있는 기능입니다.

Memcached 버전 1.4.5

Memcached 버전 1.4.5는 Amazon ElastiCache for Memcached가 지원하는 초기 엔진 및 버전이었습니다.

엔진 버전 업그레이드

사용자는 캐시 클러스터를 실행하는 프로토콜 표준 소프트웨어를 ElastiCache에서 제공하는 새 버전으로 업그레이드할지 여부와 그 시기를 조정할 수 있습니다. 이 제어 수준을 사용하면 특정 버전과의 호환성을 유지하고, 프로덕션에 배포하기 전에 애플리케이션으로 새 버전을 테스트하고, 원하는 조건과 일정에 맞춰 버전 업그레이드를 수행할 수 있습니다.

버전 업그레이드에는 약간의 호환성 위험이 있을 수 있으므로 업그레이드가 자동으로 이루어지지 않기 때문에 업그레이드는 사용자가 시작해야 합니다.

최신 Memcached 버전으로 업그레이드하려면 사용하려는 새 엔진 버전을 지정하여 캐시 클러스터를 수정합니다. 최신 Memcached 버전으로의 업그레이드는 안전하지 않은 프로세스로-, 데이터가 손상되고 콜드 캐시로 시작합니다. 자세한 내용은 [ElastiCache 클러스터 수정 \(p. 76\)](#) 단원을 참조하십시오.

이전 Memcached 버전을 Memcached 버전 1.4.33 이후로 업그레이드할 때 다음과 같은 요구 사항을 주의해야 합니다. 다음 조건에서는 CreateCacheCluster 및 ModifyCacheCluster에 실패합니다.

- $slab_chunk_max > max_item_size$ 의 경우
- $max_item_size \bmod slab_chunk_max \neq 0$ 의 경우
- $max_item_size > ((max_cache_memory - memcached_connections_overhead) / 4)$ 의 경우.

$(max_cache_memory - memcached_connections_overhead)$ 값은 데이터에 사용할 수 있는 노드의 메모리입니다. 자세한 내용은 [Memcached 연결 오버헤드 \(p. 136\)](#) 단원을 참조하십시오.

Important

- 최신 엔진 버전으로 업그레이드할 수 있지만 이전 엔진 버전으로 다운그레이드할 수 없습니다. 이전 엔진 버전을 사용하려면, 기존 클러스터를 삭제하고 이전 엔진 버전을 통해 새로 생성해야 합니다.

- 엔진 버전 관리는 패치 발생 방법을 최대한 제어할 수 있도록 설계되었습니다. 그러나 ElastiCache는 시스템 또는 캐시 소프트웨어에 심각한 보안 취약성이 발견되는 등 발생할 가능성이 거의 없는 이벤트의 경우 사용자를 대신하여 클러스터에 패치를 적용할 수 있는 권한을 보유합니다.
- Redis 엔진 버전 5.0.5부터는 가동 중지 시간을 최소화하면서 클러스터 버전을 업그레이드할 수 있습니다. 전체 업그레이드 과정 중에도 클러스터를 읽을 수 있으며, 몇 초 정도 시간이 걸리는 장애 조치 작업 중인 경우를 제외하면 대부분 업그레이드 기간 중에 쓰기도 가능합니다.
- 5.0.5 이전 버전으로 ElastiCache 클러스터를 업그레이드할 수도 있습니다. 관련된 프로세스는 동일하지만 DNS 전파 중에 장애 조치 시간이 더 길어질 수 있습니다(30s-1m).
- Memcached 엔진은 지속성을 지원하지 않으므로 Memcached 엔진 버전 업그레이드는 항상 클러스터에서 모든 캐시 데이터를 지우는 방해가 되는 프로세스입니다.

엔진 버전 업그레이드 방법

클러스터로 버전 업그레이드를 시작하려면 이를 수정하고 새 엔진 버전을 지정합니다. ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 이 작업을 수행할 수 있습니다.

- AWS Management 콘솔을 사용하려면 [AWS Management 콘솔 사용 \(p. 76\)](#)을 참조하십시오.
- AWS CLI를 사용하려면 [AWS CLI 사용 \(p. 76\)](#)을 참조하십시오.
- ElastiCache API를 사용하려면 [ElastiCache API 사용 \(p. 77\)](#)을 참조하십시오.

리전 및 가용 영역 선택

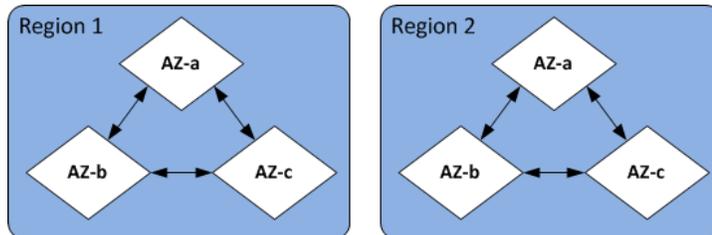
AWS 클라우드 컴퓨팅 리소스는 가용성이 매우 높은 데이터 설비에 있습니다. 추가 확장성 및 안정성을 제공하기 위해 이러한 데이터 센터 시설은 여러 물리적 위치에 배치됩니다. 이러한 위치는 지역 및 가용 영역으로 분류됩니다.

AWS 리전은 크고 광범위하게 별도의 지리적 위치에 분산되어 있습니다. 가용 영역은 다른 가용 영역에서 발생한 장애로부터 격리할 수 있도록 설계된 AWS 리전 내 개별적인 위치입니다. 가용 영역은 같은 AWS 리전에 있는 다른 가용 영역에 대해 저렴하고 지연 시간이 짧은 네트워크 연결을 제공합니다.

Important

각 리전은 완전히 독립적입니다. 시작하는 모든 ElastiCache 활동(예: 클러스터 생성)은 현재 기본 리전에서만 실행됩니다.

특정 리전에서 클러스터를 생성하거나 사용하려면 해당하는 리전 서비스 엔드포인트를 사용하십시오. 서비스 엔드포인트는 [지원되는 리전 및 엔드포인트 \(p. 31\)](#) 단원을 참조하십시오.



리전 및 가용 영역

주제

- [지원되는 리전 및 엔드포인트 \(p. 31\)](#)
- [노드 찾기 \(p. 33\)](#)

지원되는 리전 및 엔드포인트

Amazon ElastiCache는 여러 AWS 리전에서 사용할 수 있습니다. 따라서 요구 사항에 적합한 위치에서 ElastiCache 클러스터를 시작할 수 있습니다. 예를 들어, 고객과 가장 가까운 AWS 리전 또는 특정 법적 요구 사항을 준수하는 AWS 리전에서 시작할 수 있습니다.

기본적으로 AWS SDK, AWS CLI, ElastiCache API 및 ElastiCache 콘솔은 미국 서부(오레곤) 리전을 참조합니다. ElastiCache를 새 리전에서 사용할 수 있게 됨에 따라 이러한 리전의 새 엔드포인트 또한 고유 HTTP 요청, AWS SDK, AWS CLI 및 콘솔에서 사용할 수 있습니다.

각 리전은 다른 리전에서 완전히 격리되도록 설계되었습니다. 각 리전 안에는 가용 영역(AZ)이 여러 개 있습니다. 서로 다른 가용 영역에서 노드를 시작하면 가능한 최고 수준의 내결함성을 갖출 수 있습니다. 리전 및 가용 영역에 대한 자세한 내용은 이 주제의 맨 위에 있는 [리전 및 가용 영역 선택 \(p. 30\)](#) 단원을 참조하십시오.

ElastiCache가 지원되는 리전

리전 이름/리전	엔드포인트	프로토콜	
미국 동부(오하이오) 리전 us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS	
미국 동부(버지니아 북부) 지역 us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS	
미국 서부(캘리포니아 북부) 리전 us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS	
미국 서부(오레곤) 지역 us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS	
캐나다(중부) 리전 ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS	
아시아 태평양(뭄바이) 리전 ap-south-1	elasticache.ap-south-1.amazonaws.com	HTTPS	
아시아 태평양(도쿄) 리전 ap-northeast-1	elasticache.ap-northeast-1.amazonaws.com	HTTPS	
아시아 태평양(서울) 리전 ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS	
아시아 태평양(오사카-로컬) 리전 * ap-northeast-3	elasticache.ap-northeast-3.amazonaws.com	HTTPS	

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
지원되는 리전 및 엔드포인트

리전 이름/리전	엔드포인트	프로토콜	
아시아 태평양(싱가포르) 리전 ap-southeast-1	elasticache.ap-southeast-1.amazonaws.com	HTTPS	
아시아 태평양(시드니) 리전 ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS	
유럽(프랑크푸르트) 리전 eu-central-1	elasticache.eu-central-1.amazonaws.com	HTTPS	
유럽(아일랜드) 리전 eu-west-1	elasticache.eu-west-1.amazonaws.com	HTTPS	
유럽(런던) 리전 eu-west-2	elasticache.eu-west-2.amazonaws.com	HTTPS	
 eu-west-3	elasticache.eu-west-3.amazonaws.com	HTTPS	
남아메리카(상파울루) 리전 sa-east-1	elasticache.sa-east-1.amazonaws.com	HTTPS	
중국(북경) 지역 cn-north-1	elasticache.cn-north-1.amazonaws.com.cn	HTTPS	
중국(닝샤) 리전 cn-northwest-1	elasticache.cn-northwest-1.amazonaws.com.cn	HTTPS	
아시아 태평양(홍콩) 리전 ap-east-1	elasticache.ap-east-1.amazonaws.com	HTTPS	
AWS GovCloud(미국 서부) us-gov-west-1	elasticache.us-gov-west-1.amazonaws.com	HTTPS	
ElastiCache에서 AWS GovCloud(US) 사용 방법에 대한 자세한 내용은 AWS GovCloud(US) 리전의 서비스: ElastiCache 를 참조하십시오.			
참고: 아시아 태평양(오사카-로컬) 리전은 액세스를 요청하는 AWS 고객을 선택하는데 사용할 수 있는 로컬 리전입니다. 아시아 태평양(오사카-로컬) 리전을 사용하려는 경우 해당 영업 담당자에게 문의하십시오. 아시아 태평양(오사카-로컬) 리전은 단일 가용 영역을 지원합니다.			

일부 리전은 노드 유형의 하위 집합을 지원합니다. AWS 리전별로 지원되는 노드 유형을 정리한 표는 [AWS 리전별로 지원되는 노드 유형 \(p. 38\)](#) 단원을 참조하십시오.

리전별 AWS 제품 및 서비스 표는 [리전별 제품 및 서비스](#)에서 참조하십시오.

노드 찾기

Amazon ElastiCache는 AZ(가용 영역) 하나 또는 여러 개에서 클러스터의 모든 노드를 찾도록 지원합니다. 뿐만 아니라 여러 AZ에서 클러스터의 노드를 찾도록 선택하면(권장 사항) ElastiCache를 통해 각 노드의 AZ를 선택하거나 ElastiCache에서 자동으로 선택하도록 할 수 있습니다.

여러 AZ에서 노드를 찾으면 AZ 하나에서 정전과 같은 장애가 발생할 경우 전체 시스템이 실패하는 경우가 없어집니다. AZ 하나에서 모든 노드를 찾거나 여러 AZ에 노드를 분산시키는 사이에 지연 시간 차이가 크게 발생하지 않는다는 사실이 테스트를 통해 드러났습니다.

기존 클러스터를 수정할 때 노드를 추가하거나 클러스터를 생성할 때 각 노드에 대한 AZ를 지정할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- [클러스터 생성 \(p. 44\)](#)
- [ElastiCache 클러스터 수정 \(p. 76\)](#)
- [클러스터에 노드 추가 \(p. 79\)](#)

유지 관리 관리 중

모든 클러스터에는 시스템 변경 내용이 적용되는 주 단위 유지 관리 기간이 있습니다. 캐시 클러스터 생성 또는 수정할 때 원하는 유지 관리 기간을 지정하지 않으면 ElastiCache가 임의로 선택한 요일에 리전의 유지 관리 기간 내에서 60분의 유지 관리 기간을 지정합니다.

리전별로 8시간 블록 시간 중에서 60분 유지 관리 시간이 임의로 선택됩니다. 다음 표는 기본 유지 관리 기간이 할당된 각 지역별 시간 블록 목록입니다. 리전의 유지 관리 기간 블록 외부에서 원하는 유지 관리 기간을 선택할 수 있습니다.

지역 코드	리전 이름	리전 유지 관리 기간
ap-northeast-1	아시아 태평양(도쿄) 리전	13:00–21:00 UTC
ap-northeast-2	아시아 태평양(서울) 리전	12:00–20:00 UTC
ap-northeast-3	아시아 태평양(오사카-로컬) 리전	12:00–20:00 UTC
ap-south-1	아시아 태평양(뭄바이) 리전	17:30–1:30 UTC
ap-southeast-1	아시아 태평양(싱가포르) 리전	14:00–22:00 UTC
ap-southeast-2	아시아 태평양(시드니) 리전	12:00–20:00 UTC
cn-north-1	중국(베이징) 리전	14:00–22:00 UTC
eu-central-1	유럽(프랑크푸르트) 리전	23:00–07:00 UTC
eu-west-1	유럽(아일랜드) 리전	22:00–06:00 UTC
eu-west-2	유럽(런던) 리전	23:00–07:00 UTC
sa-east-1	남아메리카(상파울루) 리전	01:00–09:00 UTC
us-east-1	미국 동부(버지니아 북부) 지역	03:00–11:00 UTC

지역 코드	리전 이름	리전 유지 관리 기간
us-east-2	미국 동부(오하이오) 리전	04:00–12:00 UTC
us-gov-west-1	AWS GovCloud (US) 리전	06:00–14:00 UTC
us-west-1	미국 서부(캘리포니아 북부) 리전	06:00–14:00 UTC
us-west-2	미국 서부(오레곤) 지역	06:00–14:00 UTC

클러스터의 유지 관리 기간 변경

유지 관리 기간은 사용률이 가장 낮은 시간에 할당되어야 하므로 수시로 수정되어야 할 수 있습니다. 클러스터를 수정하여 요청한 유지 관리 활동이 이루어지는 기간을 최대 24시간까지 지정할 수 있습니다. 이 시간 동안 사용자가 요청한 지연된 또는 대기 중인 클러스터 수정이 발생합니다.

추가 정보

유지 관리 기간 및 노드 대체에 대한 자세한 내용은 다음을 참조하십시오.

- 유지 관리 및 노드 교체에 대한 [ElastiCache 유지 관리—FAQ](#)
- [노드 교체 \(p. 42\)](#)—노드 교체 관리
- [ElastiCache 클러스터 수정 \(p. 76\)](#)—클러스터의 유지 관리 기간 변경

노드 관리

노드는 Amazon ElastiCache 배포의 가장 작은 구성 요소입니다. 안전한 네트워크 부착 RAM의 크기가 고정된 청크입니다. 각 노드는 클러스터가 생성되거나 마지막으로 수정되었을 때 선택한 엔진을 실행합니다. 각 노드에는 고유한 DNS(Domain Name Service) 이름 및 포트가 있습니다. 여러 유형의 ElastiCache 노드가 지원되며, 연결된 메모리 양과 컴퓨팅 파워는 각각 다릅니다.

일반적으로 샤딩 지원으로 인해 Memcached 배포에는 더 작은 노드가 다수 있습니다. 사용할 노드 크기에 대한 자세한 내용은 [Memcached 노드 크기 선택 \(p. 68\)](#) 단원을 참조하십시오.

주제

- [노드에 연결 \(p. 34\)](#)
- [ElastiCache 예약 노드 \(p. 36\)](#)
- [지원되는 노드 유형 \(p. 36\)](#)
- [노드 교체 \(p. 42\)](#)

노드와 관련된 몇 가지 중요한 작업은 다음과 같습니다.

- [클러스터에 노드 추가 \(p. 79\)](#)
- [클러스터에서 노드 제거 \(p. 83\)](#)
- [ElastiCache for Memcached 클러스터 조정 \(p. 113\)](#)
- [연결 엔드포인트 찾기 \(p. 108\)](#)
- [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#)

노드에 연결

Memcached 클러스터에 연결하기 전에 노드의 엔드포인트가 있어야 합니다. 엔드포인트를 찾으려면 다음을 참조하십시오.

- [클러스터의 엔드포인트 찾기 \(콘솔\) \(p. 109\)](#)
- [엔드포인트 찾기 \(AWS CLI\) \(p. 110\)](#)
- [엔드포인트 찾기 \(ElastiCache API\) \(p. 112\)](#)

다음 예제에서 telnet 유틸리티를 사용하여 Memcached를 실행하는 노드에 연결합니다.

Note

Memcached 및 사용 가능한 Memcached 명령에 대한 자세한 내용은 [Memcached](#) 웹 사이트를 참조하십시오.

telnet을 사용하여 노드에 연결하려면

1. 선택한 연결 유틸리티를 사용하여 Amazon EC2 인스턴스에 연결하십시오.

Note

Amazon EC2 인스턴스에 연결하는 방법에 대한 지침은 [Amazon EC2 Getting Started Guide](#) 단원을 참조하십시오.

2. telnet 유틸리티를 다운로드하여 Amazon EC2 인스턴스에 설치합니다. Amazon EC2 인스턴스 명령 프롬프트에서 다음 명령을 입력하고 명령 프롬프트에 y를 입력합니다.

```
sudo yum install telnet
```

다음과 같이 유사한 출력이 나타납니다.

```
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. Amazon EC2 인스턴스의 명령 프롬프트에서 다음 명령을 입력하고 노드의 엔드포인트를 이 예제에 표시된 것으로 대체합니다.

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

다음과 같이 유사한 출력이 나타납니다.

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

4. Memcached 명령을 실행하여 연결을 테스트합니다.

이제 노드에 연결되어 Memcached 명령을 실행할 수 있습니다. 다음은 예제입니다.

```
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
>
```

ElastiCache 예약 노드

하나 이상의 노드를 예약하여 비용을 줄일 수 있습니다. 노드 유형과 예약 기간—(1년 또는 3년)에 따라 예약 노드에 선결제 요금이 부과됩니다. 선결제 요금 외에도 시간당 사용 요금이 부과됩니다. 이 요금은 온디맨드 노드에서 발생하는 시간당 사용 요금보다 훨씬 낮습니다.

예약된 노드가 사용 사례에 대해 비용이 절감되는지 확인하려면 먼저 필요한 노드 수와 노드 크기를 결정합니다. 그런 다음 노드의 사용량을 예측하고 온디맨드 노드와 예약된 노드의 총 비용을 비교합니다. 클러스터에서 예약 노드와 온디맨드 노드를 함께 사용할 수 있습니다. 요금 정보는 [Amazon ElastiCache 요금](#)을 참조하십시오.

자세한 내용은 [예약 노드를 통해 비용 관리](#) (p. 159) 단원을 참조하십시오.

지원되는 노드 유형

ElastiCache는 다음 노드 유형을 지원합니다. 일반적으로, 현재 세대 유형은 이전 세대의 동급 제품에 비해 더 많은 메모리와 컴퓨팅 파워를 더 저렴하게 제공합니다.

- 범용:

- 현재 세대:

- M5 노드 유형: cache.m5.large, cache.m5.xlarge, cache.m5.2xlarge, cache.m5.4xlarge, cache.m5.12xlarge, cache.m5.24xlarge

- M4 노드 유형: cache.m4.large, cache.m4.xlarge, cache.m4.2xlarge, cache.m4.4xlarge, cache.m4.10xlarge

- T3 노드 유형: cache.t3.micro, cache.t3.small, cache.t3.medium

- T2 노드 유형: cache.t2.micro, cache.t2.small, cache.t2.medium

- 이전 세대: (권장하지 않음)

- T1 노드 유형: cache.t1.micro

- M1 노드 유형: cache.m1.small, cache.m1.medium, cache.m1.large, cache.m1.xlarge

- M3 노드 유형: cache.m3.medium, cache.m3.large, cache.m3.xlarge, cache.m3.2xlarge

- 컴퓨팅 최적화:

- 이전 세대: (권장하지 않음)

- C1 노드 유형: cache.c1.xlarge

- 메모리 최적화:

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
지원되는 노드 유형

• 현재 세대:

R5 노드 유형: `cache.r5.large`, `cache.r5.xlarge`, `cache.r5.2xlarge`, `cache.r5.4xlarge`, `cache.r5.12xlarge`, `cache.r5.24xlarge`

R4 노드 유형: `cache.r4.large`, `cache.r4.xlarge`, `cache.r4.2xlarge`, `cache.r4.4xlarge`, `cache.r4.8xlarge`, `cache.r4.16xlarge`

• 이전 세대: (권장하지 않음)

M2 노드 유형: `cache.m2.xlarge`, `cache.m2.2xlarge`, `cache.m2.4xlarge`

R3 노드 유형: `cache.r3.large`, `cache.r3.xlarge`, `cache.r3.2xlarge`, `cache.r3.4xlarge`, `cache.r3.8xlarge`

Amazon ElastiCache에서 범용 순간 확장 가능 T3 표준 캐시 노드를 실행할 수 있습니다. 이러한 노드는 기존 수준의 CPU 성능과 더불어 누적된 크레딧이 소진될 때까지 언제든지 CPU 사용량을 순간 확장할 수 있는 기능을 제공합니다. CPU 크레딧은 1분 동안 CPU 코어의 전체 성능을 제공합니다.

Amazon ElastiCache의 T3 노드는 표준으로 구성되고 평균 CPU 사용률이 인스턴스의 기존 성능보다 일관되게 낮은 워크로드에 적합합니다. 기존 이상으로 순간 확장하려면 노드는 CPU 크레딧 밸런스에 누적된 크레딧을 사용합니다. 누적된 크레딧에서 노드가 부족한 경우, 성능이 점진적으로 기존 성능 수준으로 저하됩니다. 이렇게 점진적으로 저하되면 누적된 CPU 크레딧 밸런스가 고갈될 때 노드에 급격한 성능 저하가 발생하지 않습니다. 자세한 내용은 Amazon EC2 사용 설명서에서 [성능 순간 확장 가능 인스턴스에 대한 CPU 크레딧 및 기존 성능](#)을 참조하십시오.

다음 표에는 성능 순간 확장 가능 노드 유형과 시간당 CPU 크레딧이 획득되는 속도가 나와 있습니다. 또한 노드가 누적할 수 있는 최대 획득 CPU 크레딧 개수와 노드당 vCPU 개수도 보여줍니다. 또한 기존 성능 수준을 전체 코어 성능의 백분율로 제공합니다(단일 vCPU 사용).

시간당 획득된 CPU 크레딧	누적 가능한 최대 획득된 크레딧*	vCPUs	vCPU당 기존 성능	메모리(GiB)	네트워크 성능
t3 micro	288	2	10%	0.5	최대 5 기가비트
t3 small	576	2	20%	1.37	최대 5 기가비트
t3 medium	576	2	20%	3.09	최대 5 기가비트

* 누적될 수 있는 크레딧은 수는 24시간 동안 획득할 수 있는 크레딧의 수와 동일합니다.

** 테이블의 기존 성능은 vCPU 단위로 계산됩니다. vCPU가 1개 이상인 일부 노드 크기입니다. vCPU 백분율에 vCPU 개수를 곱하여 노드의 기존 CPU 사용률을 계산합니다.

다음 CPU 크레딧 지표는 성능 순간 확장 가능 인스턴스에 사용할 수 있습니다.

- CPUCreditUsage
- CPUCreditBalance

이 지표에 대한 자세한 내용은 [CPU 크레딧 지표](#) 단원을 참조하십시오.

또한 다음 사항을 숙지해야 합니다.

- 모든 현재 세대 노드 유형은 기본적으로 Amazon VPC를 기반으로 Virtual Private Cloud(VPC)에서 생성됩니다.

Note

지원되는 엔진 버전은 AWS 리전별로 다릅니다. 최신 엔진 버전은 모든 AWS 리전에서 지원됩니다. AWS 리전에 사용 가능한 엔진 버전을 확인하려면 [지원되는 ElastiCache for Memcached 버전 \(p. 27\)](#) 단원을 참조하십시오.

AWS 리전별로 지원되는 노드 유형

다음 표에는 각 AWS 리전에 지원되는 노드 유형이 나와 있습니다.

AWS 리전	T3	T2	M4	M5	R4	R5
us-east-2 미국 북부 부 (오 하이 오)	예	예	예	예	예	예
us-east-1 미국 북부 부 (버 지니 아 북 부)	예	예	예	예	예	예
us-west-1 미국 서부 부 (캘 리 포 니 아 북 부 지 역)	예	예	예	예	예	예
us-west-2 미국 서부 부 (오 리 곤)	예	예	예	예	예	예

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
지원되는 노드 유형

AWS 리전	T3	T2	M4	M5	R4	R5
레곤)						
ca-central-1 나 다 (중 부)	예	예	예	예	예	예
ap-south-1 시 아 태 평양 (뭄 바 이)	예	예	예	예	예	예
ap-northeast-1 시 아 태 평양 (도 쿄)	예	예	예	예	예	예
ap-northeast-2 시 아 태 평양 (서 울)	예	예	예	예	예	예
ap-northeast-3 시 아 태 평양 (오 사 카- 로 컬) *	예	예	예	예	예	

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
지원되는 노드 유형

AWS 리전	T3	T2	M4	M5	R4	R5
ap-southeast-1 아시아 태평양 (싱가포르)	예	예	예	예	예	예
ap-southeast-2 아시아 태평양 (시드니)	예	예	예	예	예	예
ap-east-1 아시아 태평양 (홍콩)	예			예		예
eu-north-1 유럽 (스톡홀름)	예			예	아니오	예
eu-central-1 유럽 (프랑크푸르트)	예	예	예	예	예	예
eu-west-1 유럽 (아일랜드)	예	예	예	예	예	예

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
지원되는 노드 유형

AWS 리전	T3	T2	M4	M5	R4	R5
eu-west-2 유럽 (런던)	예	예	예	예	예	예
eu-west-3 유럽 (프랑크푸르트)	예	예		예	예	예
na-east-1 북아메리카 (상파울루)	예	예	예	예	예	예
cn-north-1 중국 (베이징)	예	예	예		예	예
cn-northwest-1 중국 (닝샤)	예	예	예		예	예
me-south-1 중동 (바레인)	예			예		예
aws-gov-west-1 AWS GovCloud (US-West)	예	예		예	예	예
* 아시아 태평양(오사카-로컬) 리전은 액세스를 요청하는 AWS 고객을 선택하는 데 사용할 수 있는 로컬 리전입니다. 아시아 태평양(오사카-로컬) 리전을 사용하려는 경우 해당 영업 담당자에게 문의하십시오. 아시아 태평양(오사카-로컬) 리전은 단일 가용 영역을 지원합니다.						

노드 유형 및 사양의 전체 목록은 다음을 참조하십시오.

- [Amazon ElastiCache 제품 기능 및 세부 정보](#)
- [Memcached 노드 유형별 파라미터](#)

노드 교체

Amazon ElastiCache for Memcached는 인스턴스에 원활하게 적용되는 패치 및 업그레이드를 통해 플릿을 주기적으로 업그레이드합니다. 하지만 기본 호스트에 필수 OS 업데이트를 적용하기 위해 ElastiCache for Memcached 노드를 다시 시작해야 하는 경우가 있습니다. 보안, 안정성 및 운영 성능을 강화하는 업그레이드 적용에 있어 이러한 교체가 필요합니다.

예정된 노드 교체 주기 이전에 언제든지 이러한 교체를 직접 관리할 수 있는 옵션이 있습니다. 직접 대체를 관리할 때 노드를 다시 시작하면 인스턴스에서 OS 업데이트를 수신하고, 예정된 노드 대체는 취소됩니다. 노드 대체가 발생한다는 경고를 계속 수신할 수 있습니다. 이미 유지 관리의 필요성을 수동으로 완화한 경우 이 경고를 무시할 수 있습니다.

Note

Amazon ElastiCache에서 자동으로 생성된 교체 캐시 노드는 IP 주소가 다를 수 있습니다. 애플리케이션 구성을 검토하여 캐시 노드가 적절한 IP 주소와 연결되어 있는지 확인해야 합니다.

다음 목록은 ElastiCache에서 Memcached 노드 하나의 대체를 예약할 경우 취할 수 있는 조치를 보여줍니다.

- 아무 작업 안 함 - 아무 작업도 하지 않으면 ElastiCache에서 예약대로 노드를 대체합니다. ElastiCache에서 자동으로 노드를 새 노드로 대체하면 새 노드가 처음에는 비어 있습니다.
- 유지 관리 기간 변경 - 예약된 유지 관리 이벤트의 경우 ElastiCache에서 이메일 또는 알림 이벤트를 수신합니다. 이 경우 예약된 대체 시간 전에 유지 관리 기간을 변경하면 이제 노드가 새 시간에 대체됩니다. 자세한 내용은 [ElastiCache 클러스터 수정 \(p. 76\)](#) 단원을 참조하십시오.

Note

유지 관리 기간을 이동해 교체 기간을 변경하는 기능은 ElastiCache 알림에 유지 관리 기간이 포함된 경우에만 사용할 수 있습니다. 알림에 유지 관리 기간이 포함되어 있지 않으면 교체 기간을 변경할 수 없습니다.

예를 들어 11월 9일 목요일 15:00, 다음 유지 관리 기간은 11월 10일 금요일 17:00라고 가정해 보겠습니다. 다음을 이러한 가정의 결과를 보여주는 3가지 시나리오입니다.

- 유지 관리 기간을 현재 날짜/시간 이후 및 예약된 다음 유지 관리 기간 이전인 금요일 16:00으로 변경합니다. 11월 10일 금요일 16:00에 노드가 대체됩니다.
- 유지 관리 기간을 현재 날짜/시간 이후 및 예약된 다음 유지 관리 기간 이전인 토요일 16:00으로 변경합니다. 11월 11일 토요일 16:00에 노드가 대체됩니다.
- 유지 관리 기간을 현재 날짜/시간보다 일주일 빠른 수요일 오후 4시로 변경합니다. 11월 15일 수요일 16:00에 노드가 대체됩니다.

지침은 [유지 관리 관리 중 \(p. 33\)](#) 단원을 참조하십시오.

- 수동으로 노드 대체 - 다음 유지 관리 기간 전에 노드를 대체하려면 노드를 수동으로 대체하십시오.

노드를 수동으로 대체하면 키는 재분산됩니다. 이 재분산으로 인해 캐시가 누락될 수 있습니다.

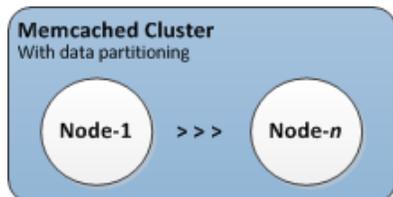
Memcached 노드를 수동으로 대체하려면

1. 대체 예약한 노드를 삭제합니다. 지침은 [클러스터에서 노드 제거 \(p. 83\)](#) 단원을 참조하십시오.
2. 클러스터에 새 노드를 추가합니다. 지침은 [클러스터에 노드 추가 \(p. 79\)](#) 단원을 참조하십시오.
3. 이 클러스터에서 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 기능을 사용하지 않으려면 애플리케이션을 확인하고 기존 노드의 모든 엔드포인트 인스턴스를 새로운 노드 엔드포인트로 대체합니다.

ElastiCache 클러스터 관리

클러스터는 하나 이상의 캐시 노드 모음으로서, 이 모든 노드는 Memcached 캐시 엔진 소프트웨어의 인스턴스 하나를 실행합니다. 클러스터를 만들 때 모든 노드에서 사용할 엔진과 버전을 지정합니다.

다음 다이어그램은 일반적인 Memcached 클러스터를 나타낸 것입니다. Memcached 클러스터에는 데이터를 가로로 분할할 수 있는 노드가 1 ~ 20개까지 포함됩니다.



일반적인 Memcached 클러스터

클러스터 수준에서 대부분의 ElastiCache 작업이 수행됩니다. 특정 수의 노드 및 각 노드에 대한 속성을 제어하는 파라미터 그룹을 사용하여 클러스터를 설정할 수 있습니다. 클러스터 하나에 속한 모든 노드는 노드 유형, 파라미터 및 보안 그룹 설정이 동일합니다.

클러스터마다 클러스터 식별자가 있습니다. 클러스터 식별자는 고객이 제공하는 클러스터 이름입니다. ElastiCache API 및 AWS CLI 명령과 상호 작용할 때 이 식별자가 특정한 클러스터를 지정합니다. 클러스터 식별자는 한 AWS 리전 내의 해당 고객에 대해 고유해야 합니다.

ElastiCache는 여러 가지 엔진 버전을 지원합니다. 특별한 이유가 없으면 항상 엔진의 최신 버전을 사용하는 것이 좋습니다.

ElastiCache 클러스터는 Amazon EC2 인스턴스를 사용하여 액세스하도록 설계되었습니다. Amazon VPC 서비스 기반의 VPC에서 클러스터를 시작하는 경우 AWS 밖에서 액세스할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- [2단계: 액세스 자격 부여 \(p. 16\)](#)
- [AWS 외부에서 ElastiCache 리소스 액세스 \(p. 93\)](#)

지원되는 Memcached 버전

- [Memcached 버전 1.4.34 \(p. 28\)](#)
- [Memcached 버전 1.4.33 \(p. 28\)](#)
- [Memcached 버전 1.4.24 \(p. 29\)](#)
- [Memcached 버전 1.4.14 \(p. 29\)](#)
- [Memcached 버전 1.4.5 \(p. 29\)](#)

기타 ElastiCache 클러스터 작업

클러스터와 관련된 추가 작업:

- [연결 엔드포인트 찾기 \(p. 108\)](#)
- [AWS 외부에서 ElastiCache 리소스 액세스 \(p. 93\)](#)

클러스터 생성

이 단원에는 ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하는 클러스터 생성에 관한 지침이 나와 있습니다.

ElastiCache 클러스터는 [AWS CloudFormation](#)을 사용하여 생성할 수도 있습니다. 자세한 내용은 해당 접근 방식을 구현하는 방법에 대한 지침이 포함되어 있는 [AWS CloudFormation 사용 설명서](#)에서 [AWS::ElastiCache::CacheCluster](#)를 참조하십시오.

클러스터를 생성할 때마다 준비 작업을 미리 하면 즉시 업그레이드하거나 변경할 필요가 없어 좋습니다.

주제

- [요구 사항 결정 \(p. 44\)](#)
- [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#)
- [노드 크기 선택 \(p. 68\)](#)
- [Memcached 클러스터 생성 \(콘솔\) \(p. 69\)](#)
- [클러스터 생성 \(AWS CLI\) \(p. 71\)](#)
- [클러스터 생성 \(ElastiCache API\) \(p. 72\)](#)

요구 사항 결정

주제

- [메모리 및 프로세서 요구 사항 \(p. 45\)](#)
- [Memcached 클러스터 구성 \(p. 45\)](#)
- [조정 요구 사항 \(p. 45\)](#)
- [액세스 요구 사항 \(p. 45\)](#)
- [리전 및 가용 영역 요구 사항 \(p. 46\)](#)

준비

시작하기 전에 다음 질문에 대한 답을 알아야 클러스터를 생성할 수 있습니다.

- 필요한 노드 인스턴스 유형은 무엇입니까?

인스턴스 노드 유형 선택에 도움이 필요한 경우 [Memcached 노드 크기 선택 \(p. 68\)](#)을 참조하십시오.

- VPC와 Amazon VPC 중 어디서 클러스터를 시작하시겠습니까?

Important

Amazon VPC에서 클러스터를 시작하는 경우 클러스터를 생성하기 전에 동일한 VPC에서 서브넷 그룹을 생성해야 합니다. 자세한 내용은 [서브넷 및 서브넷 그룹 \(p. 182\)](#) 단원을 참조하십시오.

ElastiCache는 Amazon EC2를 사용하여 AWS 내에서 액세스하도록 고안되었습니다. 하지만 Amazon VPC 기반의 VPC에서 시작하고 클러스터가 VPC에 있는 경우 AWS 밖에서 액세스 권한을 제공할 수 있습니다. 자세한 내용은 [AWS 외부에서 ElastiCache 리소스 액세스 \(p. 93\)](#) 단원을 참조하십시오.

- 파라미터 값을 사용자 지정해야 합니까?

그렇다면 사용자 지정 파라미터 그룹을 생성해야 합니다. 자세한 내용은 [파라미터 그룹 생성 \(p. 116\)](#) 단원을 참조하십시오.

- 자체 보안 그룹 또는 VPC 보안 그룹을 생성해야 합니까?

자세한 내용은 [보안 그룹: EC2-Classic \(p. 188\)](#) 및 [VPC의 보안](#)을 참조하십시오.

- 어떤 방법으로 내결함성을 구현하시겠습니까?

자세한 내용은 [장애 완화 \(p. 213\)](#) 단원을 참조하십시오.

주제

- [메모리 및 프로세서 요구 사항 \(p. 45\)](#)
- [Memcached 클러스터 구성 \(p. 45\)](#)
- [조정 요구 사항 \(p. 45\)](#)
- [액세스 요구 사항 \(p. 45\)](#)
- [리전 및 가용 영역 요구 사항 \(p. 46\)](#)

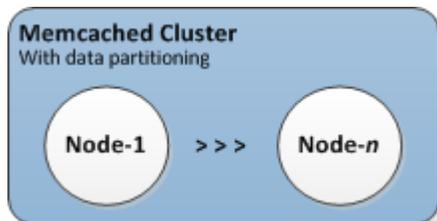
메모리 및 프로세서 요구 사항

Amazon ElastiCache의 기본 빌딩 블록은 노드입니다. 노드는 개별적으로 또는 그룹으로 구성되어 클러스터를 형성합니다. 클러스터에 사용할 노드 유형을 결정할 때 클러스터의 노드 구성과 저장해야 하는 데이터의 양을 고려합니다.

Memcached 엔진은 다중 스레드이므로 노드의 코어 수가 클러스터에 사용할 수 있는 컴퓨팅 파워에 영향을 줍니다.

Memcached 클러스터 구성

ElastiCache for Memcached 클러스터는 1개에서 20개의 노드로 구성됩니다. Memcached 클러스터의 데이터는 클러스터의 노드로 분할됩니다. 애플리케이션은 엔드포인트라는 네트워크 주소를 사용하여 Memcached 클러스터에 연결됩니다. Memcached 클러스터의 각 노드에는 애플리케이션이 특정 노드(서) 읽고 쓰는 데 사용하는 자체 엔드포인트가 있습니다. 노드 엔드포인트 외에도 Memcached 클러스터는 구성 엔드포인트라는 엔드포인트를 가지고 있습니다. 애플리케이션에서는 이 엔드포인트를 사용하여 클러스터에서 읽거나 쓸 수 있으며, [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#)에서 읽을 노드 또는 쓸 노드에 대한 결정을 남깁니다.



자세한 내용은 [ElastiCache 클러스터 관리 \(p. 43\)](#) 단원을 참조하십시오.

조정 요구 사항

더 크고 새로운 노드 유형으로 새 클러스터를 생성하여 모든 클러스터를 확장할 수 있습니다. Memcached 클러스터를 확장하면 새 클러스터가 비워집니다.

Memcached 클러스터용 Amazon ElastiCache는 확장하거나 축소할 수 있습니다. Memcached 클러스터를 스케일 아웃하거나 스케일 인하려면 클러스터에서 노드를 추가하거나 삭제하십시오. 자동 검색을 사용하고 애플리케이션이 클러스터의 구성 엔드포인트와 연결된 경우 노드를 추가하거나 제거할 때 애플리케이션을 변경할 필요가 없습니다.

자세한 내용은 이 가이드의 [ElastiCache for Memcached 클러스터 조정 \(p. 113\)](#)을 참조하십시오.

액세스 요구 사항

설계에 따라 Amazon ElastiCache 클러스터는 Amazon EC2 인스턴스에서 액세스합니다. ElastiCache 클러스터에 대한 네트워크 액세스는 클러스터를 생성한 사용자 계정에 대해 제한됩니다. 따라서 Amazon EC2 인

스턴스에서 클러스터에 액세스하려면 먼저 Amazon EC2 인스턴스가 클러스터에 액세스하도록 승인해야 합니다. 이를 수행하는 단계는 EC2-VPC로 시작했는지, EC2-Classic으로 시작했는지에 따라 다릅니다.

클러스터를 EC2-VPC로 시작한 경우 클러스터에 대한 네트워크 진입을 허용해야 합니다. 클러스터를 EC2-Classic으로 시작한 경우 Amazon Elastic Compute Cloud 보안 그룹에 액세스하여 인스턴스와 연결된 ElastiCache 보안 그룹을 허용해야 합니다. 자세한 지침은 이 가이드의 [2단계: 액세스 자격 부여 \(p. 16\)](#)를 참조하십시오.

리전 및 가용 영역 요구 사항

Amazon ElastiCache은 모든 AWS 리전을 지원합니다. 애플리케이션과 가까운 리전에 ElastiCache 클러스터를 배치하면 지연 시간을 줄일 수 있습니다. 클러스터에 다중 노드가 있는 경우 다른 가용 영역에 노드를 배치하면 클러스터에 장애가 미치는 영향을 줄일 수 있습니다.

자세한 내용은 다음 자료를 참조하십시오.

- [리전 및 가용 영역 선택 \(p. 30\)](#)
- [장애 완화 \(p. 213\)](#)

Memcached 클러스터의 노드를 자동으로 식별

Memcached 엔진을 실행하는 클러스터에 대해 ElastiCache는 자동 검색—을 지원합니다. 자동 검색은 클라이언트 프로그램이 캐시 클러스터의 모든 노드를 자동으로 식별하고 이러한 모든 노드와의 연결을 시작하고 유지하는 기능입니다.

Note

Amazon ElastiCache Memcached에서 실행되는 캐시 클러스터를 위해 자동 검색이 추가되었습니다.

자동 검색을 사용하면 애플리케이션이 개별 캐시 노드에 수동으로 연결할 필요가 없으며, 대신 애플리케이션이 Memcached 노드 하나에 연결하고 노드 목록을 검색합니다. 그 목록에서 애플리케이션은 클러스터의 나머지 노드를 인식하고 이러한 모든 노드에 연결할 수 있습니다. 애플리케이션에서 개별 캐시 노드 엔드포인트를 하드 코딩할 필요가 없습니다.

클러스터에 있는 모든 캐시 노드는 모든 다른 노드에 대한 메타데이터 목록을 유지 관리합니다. 이 메타데이터는 클러스터에서 노드가 추가되거나 제거될 때마다 업데이트됩니다.

주제

- [자동 검색의 이점 \(p. 46\)](#)
- [Auto Discovery 작동 방법 \(p. 47\)](#)
- [자동 검색 사용 \(p. 49\)](#)
- [캐시 노드에 수동으로 연결 \(p. 54\)](#)
- [Auto Discovery를 클라이언트 라이브러리에 추가 \(p. 54\)](#)
- [ElastiCache 자동 검색이 있는 클라이언트 \(p. 56\)](#)

자동 검색의 이점

Auto Discovery는 다음 혜택을 제공합니다.

- 캐시 클러스터에서 노드 개수를 늘리면 새로운 노드가 구성 Endpoint 및 모든 다른 노드를 사용하여 자신을 등록합니다. 캐시 클러스터에서 노드를 제거하면 제거되는 노드가 자신을 등록 해제합니다. 두 가지 경우 모두에서 클러스터의 다른 노드 모두는 최신 캐시 노드 메타데이터를 사용하여 업데이트됩니다.

- 캐시 노드 실패가 자동으로 감지되고 실패한 노드는 자동으로 대체됩니다.

Note

노드 대체가 완료될 때까지 노드가 계속 실패합니다.

- 클라이언트 프로그램은 구성 endpoint에 연결해야만 합니다. 그 후에 자동 검색 라이브러리가 클러스터의 다른 모든 노드에 연결됩니다.
- 클라이언트 프로그램은 분당(이 간격은 필요한 경우 조정 가능) 한 번씩 클러스터를 폴링합니다. 노드가 추가 또는 삭제되는 등의 클러스터 구성에 변경 사항이 있는 경우 클라이언트는 업데이트된 메타데이터 목록을 받습니다. 그러면 클라이언트가 필요에 따라 이 노드에 연결하거나 연결을 해제합니다.

모든 ElastiCache Memcached 캐시 클러스터에서 자동 검색이 활성화됩니다. 이 기능을 사용하기 위해 캐시 노드 중 어느 것도 재부팅할 필요가 없습니다.

Auto Discovery 작동 방법

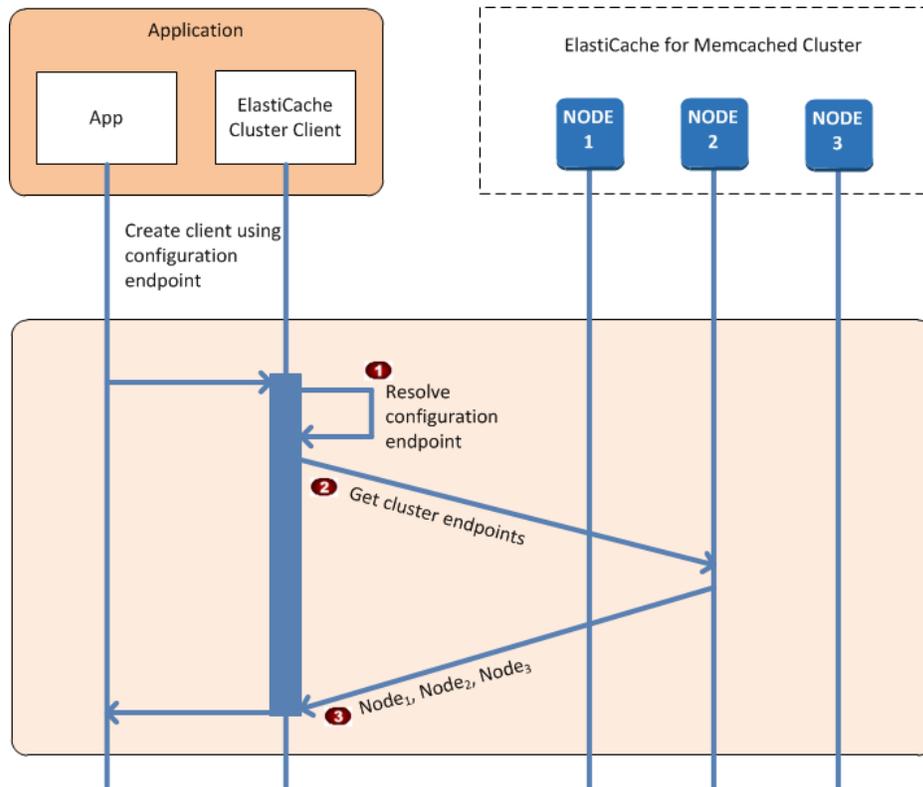
주제

- 캐시 노드에 연결 (p. 47)
- 일반적인 클러스터 작업 (p. 48)
- 기타 작업 (p. 49)

이 단원은 클라이언트 애플리케이션이 ElastiCache 클러스터 클라이언트를 사용하여 캐시 노드 연결을 관리하고 캐시의 데이터와 상호 작용하는 방법을 설명합니다.

캐시 노드에 연결

애플리케이션 관점에서 클러스터 구성 엔드포인트에 대한 연결은 개별 캐시 노드에 대한 직접 연결과 다르지 않습니다. 다음 시퀀스 다이어그램은 캐시 노드에 연결하는 과정을 보여 줍니다.



캐시 노드에 연결하는 과정

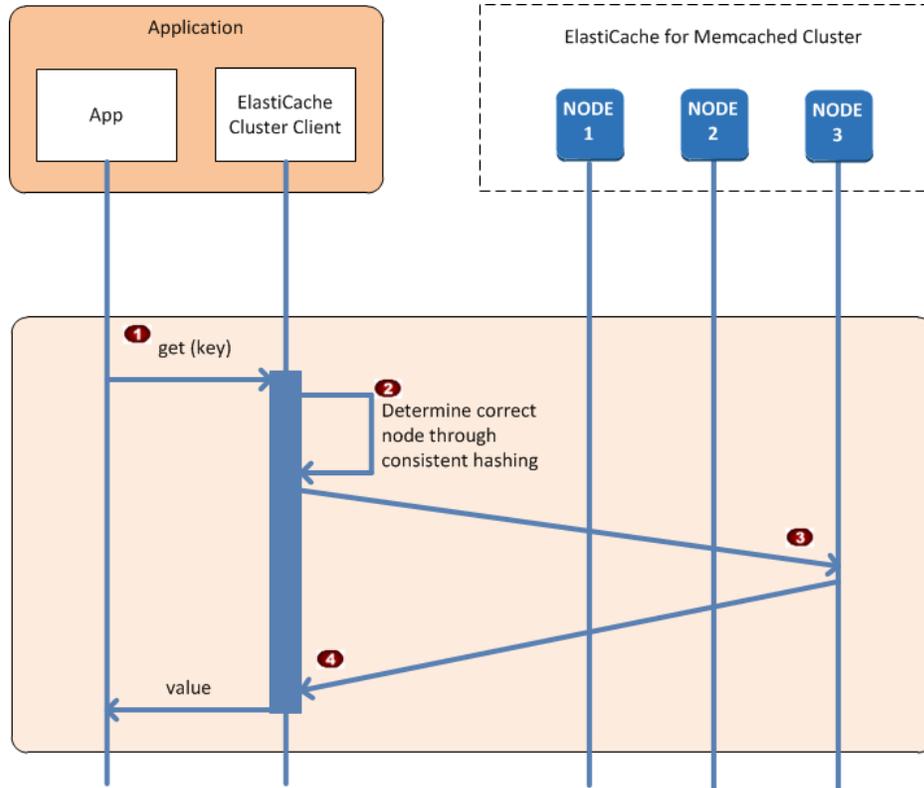
- 1 애플리케이션이 구성 Endpoint의 DNS 이름을 확인합니다. 구성 Endpoint가 모든 캐시 노드의 CNAME 항목을 유지 관리하기 때문에 DNS 이름은 노드 중 하나에 대해 확인되며 해당 클라이언트가 해당 노드에 연결합니다.
- 2 클라이언트는 모든 다른 노드에 대한 구성 정보를 요청합니다. 각 노드가 클러스터에 있는 모든 노드에 대한 구성 정보를 유지 관리하므로 요청 시 모든 노드가 구성 정보를 클라이언트에 전달할 수 있습니다.
- 3 클라이언트는 캐시 노드 호스트 이름 및 IP 주소의 현재 목록을 받습니다. 그런 다음 클러스터에 있는 모든 다른 노드에 연결할 수 있습니다.

Note

클라이언트 프로그램은 캐시 노드 호스트 이름 및 IP 주소 목록을 분당 한 번씩 새로 고칩니다. 이 폴링 간격은 필요한 경우 조정될 수 있습니다.

일반적인 클러스터 작업

애플리케이션이 모든 캐시 노드에 연결되었을 때 ElastiCache 클러스터 클라이언트는 개별 데이터 항목을 저장해야 하는 노드 및 나중에 그러한 데이터 항목에 대해 쿼리되어야 하는 노드를 결정합니다. 다음 시퀀스 다이어그램은 일반 클러스터 작업 과정을 보여 줍니다.



일반적인 클러스터 작업 과정

- 1 애플리케이션이 키로 식별되는 특정 데이터 항목에 대해 Get 요청을 실행합니다.
- 2 클라이언트가 키에 대해 해시 알고리즘을 사용하여 해당 데이터 항목을 포함하는 캐시 노드를 결정합니다.

- 3 데이터 항목이 적절한 노드에서 요청됩니다.
- 4 데이터 항목이 애플리케이션으로 반환됩니다.

기타 작업

일부 경우에는 클러스터의 노드를 변경할 수 있습니다. 예를 들어, 추가 수요를 수용하기 위해 노드를 추가하거나, 수요가 감소하는 기간 동안 비용을 절약하기 위해 노드를 삭제할 수 있습니다. 또는 이런 저런 노드 장애로 인해 노드를 대체할 수도 있습니다.

클러스터가 변경되어 클러스터 엔드포인트로 메타데이터를 업데이트해야 하는 경우 동시에 모든 노드가 변경되므로 제공된 모든 노드의 메타데이터는 클러스터에 있는 다른 모든 노드의 메타데이터와 일치합니다.

이런 경우 클러스터의 모든 노드에서 동시에 메타데이터가 업데이트되므로 메타데이터는 항상 모든 노드에서 일치합니다. 항상 구성 엔드포인트를 사용하여 클러스터에 있는 여러 노드의 엔드포인트를 가져와야 합니다. 구성 엔드포인트를 사용하여 "사라지는" 노드에서 엔드포인트 데이터를 가져오지 않을 수 있습니다.

노드 추가

노드가 실행되는 동안 노드의 엔드포인트가 메타데이터에 포함되지 않습니다. 노드가 사용 가능 상태가 되면 각 클러스터 노드의 메타데이터에 노드가 추가됩니다. 이 시나리오에서는 모든 노드에서 메타데이터가 일치하고 노드가 사용 가능해진 후에야 새로운 노드와 상호 작용할 수 있습니다. 노드가 사용 가능 상태가 되기 전에는 노드에 대해 알 수 없으며 새로운 노드가 존재하지 않는 것처럼 클러스터에 있는 노드와 상호 작용합니다.

노드 삭제

노드가 제거되면 노드의 엔드포인트가 먼저 메타데이터에서 제거된 후 클러스터에서 노드가 제거됩니다. 이 시나리오에서는 모든 노드의 메타데이터가 일치하고 노드가 사용 가능 상태가 아닐 때 제거할 노드의 엔드포인트를 포함할 시간이 없습니다. 노드 제거 시간 중에는 메타데이터에 노드가 보고되지 않으므로 애플리케이션이 노드가 존재하지 않는 것처럼 나머지 노드 n-1개에 한해 상호 작용합니다.

노드 대체

노드가 실패하면 ElastiCache가 해당 노드의 작동을 중지하고 대체합니다. 대체 프로세스는 몇 분 정도 걸립니다. 이 시간 동안 모든 노드의 메타데이터가 실패한 노드의 엔드포인트를 계속 표시하지만 이 노드와 상호 작용하려는 모든 시도가 실패합니다. 따라서 로직에 항상 재시도 로직을 포함해야 합니다.

자동 검색 사용

Auto Discovery를 사용하려면 다음 단계를 따릅니다.

- 1단계: 구성 Endpoint 확보 (p. 49)
- 2단계: ElastiCache 클러스터 클라이언트 다운로드 (p. 51)
- 3단계: 애플리케이션 프로그램 수정 (p. 51)

1단계: 구성 Endpoint 확보

클라이언트 프로그램은 클러스터에 연결하기 위해 클러스터 구성 Endpoint를 알아야 합니다. [클러스터의 엔드포인트 찾기 \(콘솔\)](#) (p. 109) 항목을 참조하십시오.

또한 다음과 같이 `--show-cache-node-info` 매개 변수를 포함하여 `aws elasticache describe-cache-clusters` 명령을 사용할 수 있습니다.

클러스터의 엔드포인트를 찾기 위해 어떤 방법을 사용하든지 구성 엔드포인트는 항상 주소에 `.cfg`를 포함합니다.

Example ElastiCache용 AWS CLI를 사용하여 엔드포인트 찾기

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

이 작업은 다음과 비슷한 출력을 생성합니다(JSON 형식).

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

2단계: ElastiCache 클러스터 클라이언트 다운로드

자동 검색을 이용하려면 클라이언트 프로그램에서 ElastiCache 클러스터 클라이언트를 사용해야 합니다. ElastiCache 클러스터 클라이언트는 Java, PHP 및 .NET에 사용할 수 있으며 모든 캐시 노드를 검색하고 연결하는 데 필요한 모든 로직을 포함합니다.

ElastiCache 클러스터 클라이언트를 다운로드하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. ElastiCache 콘솔에서 ElastiCache Cluster Client(클러스터 클라이언트)를 선택한 후 다운로드를 선택합니다.

Java용 ElastiCache 클러스터 클라이언트의 소스 코드는 <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java>에 있습니다. 이 라이브러리는 일반적으로 많이 사용되는 Spymemcached 클라이언트를 기반으로 합니다. ElastiCache 클러스터 클라이언트는 Amazon Software License <https://aws.amazon.com/ASL>에 따라 릴리스됩니다. 원하는 대로 자유롭게 소스 코드를 수정할 수 있습니다. 소스를 다른 오픈 소스 Memcached 라이브러리 또는 자체 클라이언트 코드에 통합할 수도 있습니다.

Note

PHP용 ElastiCache 클러스터 클라이언트를 사용하려면 먼저 클라이언트를 Amazon EC2 인스턴스에 설치해야 합니다. 자세한 내용은 [PHP용 ElastiCache 클러스터 클라이언트 설치 \(p. 58\)](#) 단원을 참조하십시오.

.NET용 ElastiCache 클러스터 클라이언트를 사용하려면 먼저 클라이언트를 Amazon EC2 인스턴스에 설치해야 합니다. 자세한 내용은 [.NET용 ElastiCache 클러스터 클라이언트 설치 \(p. 56\)](#) 단원을 참조하십시오.

3단계: 애플리케이션 프로그램 수정

자동 검색을 사용하도록 애플리케이션 프로그램을 수정합니다. 다음 단원에서는 Java, PHP 및 .NET용 ElastiCache를 사용하는 방법을 보여줍니다.

Important

클러스터의 구성 엔드포인트를 지정할 때 다음과 같이 엔드포인트의 주소에 ".cfg"를 포함해야 합니다. ".cfg"를 포함하지 않은 엔드포인트 또는 CNAME을 사용하지 마십시오.

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

클러스터의 구성 엔드포인트를 명시적으로 지정하지 못하면 특정 노드로 구성됩니다.

주제

- [Java용 ElastiCache 클러스터 클라이언트 사용 \(p. 52\)](#)
- [PHP용 ElastiCache 클러스터 클라이언트 사용 \(p. 52\)](#)
- [.NET용 ElastiCache 클러스터 클라이언트 사용 \(p. 53\)](#)

Java용 ElastiCache 클러스터 클라이언트 사용

아래 프로그램은 ElastiCache 클러스터 클라이언트를 사용하여 클러스터 구성 엔드포인트에 연결하고 데이터 항목을 캐시에 추가하는 방법을 설명합니다. 프로그램은 자동 검색을 사용하여 추가 개입 없이 클러스터에 있는 모든 노드에 연결합니다.

```
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");

    }
}
```

PHP용 ElastiCache 클러스터 클라이언트 사용

아래 프로그램은 ElastiCache 클러스터 클라이언트를 사용하여 클러스터 구성 엔드포인트에 연결하고 데이터 항목을 캐시에 추가하는 방법을 설명합니다. 프로그램은 Auto Discovery를 사용하여 추가 개입 없이 클러스터에 있는 모든 노드에 연결합니다.

PHP용 ElastiCache 클러스터 클라이언트를 사용하려면 먼저 클라이언트를 Amazon EC2 인스턴스에 설치해야 합니다. 자세한 내용은 [PHP용 ElastiCache 클러스터 클라이언트 설치 \(p. 58\)](#) 단원을 참조하십시오.

```
<?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
 */
```

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
클러스터 생성

```
* 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');
?>
```

.NET용 ElastiCache 클러스터 클라이언트 사용

ElastiCache용 .NET 클라이언트는 <https://github.com/awslabs/elasticache-cluster-config-net>에 있는 오픈 소스입니다.

.NET 애플리케이션은 대개 구성 파일에서 구성을 가져옵니다. 다음은 샘플 애플리케이션 구성 파일입니다.

```
<?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>
```

아래 C# 프로그램은 ElastiCache 클러스터 클라이언트를 사용하여 클러스터 구성 엔드포인트에 연결하고 데이터 항목을 캐시에 추가하는 방법을 설명합니다. 프로그램은 Auto Discovery를 사용하여 추가 개입 없이 클러스터에 있는 모든 노드에 연결합니다.

```
// *****
```

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
클러스터 생성

```
// 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
```

캐시 노드에 수동으로 연결

클라이언트 프로그램이 Auto Discovery를 사용하지 않는 경우 프로그램은 수동으로 캐시 노드 각각에 연결할 수 있습니다. 이 방법이 Memcached 클라이언트의 기본 방법입니다.

[AWS Management Console](#)에서 캐시 노드 호스트 이름 및 포트 번호 목록을 확보할 수 있습니다. 또한 다음과 같이 `--show-cache-node-info` 매개 변수를 포함하여 AWS CLI `aws elasticache describe-cache-clusters` 명령을 사용할 수 있습니다.

Example

다음 Java 코드 조각은 노드가 4개인 캐시 클러스터에 있는 모든 노드에 연결하는 방법을 보여 줍니다.

```
...

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

노드를 추가 또는 삭제하여 캐시 클러스터를 확장 또는 축소하는 경우 클라이언트 노드에서 노드 목록을 업데이트해야 합니다.

Auto Discovery를 클라이언트 라이브러리에 추가

Auto Discovery의 구성 정보는 각 캐시 클러스터 노드에 중복 저장됩니다. 클라이언트 애플리케이션은 모든 캐시 노드를 쿼리하고 클러스터에서 모든 노드의 구성 정보를 확보할 수 있습니다.

애플리케이션이 이 작업을 수행하는 방법은 캐시 엔진 버전에 따라 달라집니다.

- 캐시 엔진 버전이 1.4.14 이상인 경우 `config` 명령을 사용합니다.
- 캐시 엔진 버전이 1.4.14 미만인 경우 `get AmazonElastiCache:cluster` 명령을 사용합니다.

이러한 두 명령의 출력은 동일하며 아래 [출력 형식 \(p. 55\)](#) 단원에서 설명됩니다.

캐시 엔진 버전 1.4.14 이상

캐시 엔진 버전 1.4.14 이상인 경우 `config` 명령을 사용합니다. 이 명령은 ElastiCache에 의해 Memcached ASCII 및 바이너리 프로토콜에 추가되었으며 ElastiCache 클러스터 클라이언트에서 구현됩니다. 다른 클라이언트 라이브러리에 Auto Discovery를 사용하려면 해당 라이브러리가 `config` 명령을 지원하도록 확장될 필요가 있습니다.

Note

다음 설명은 ASCII 프로토콜에 관련된 것이지만 `config` 명령은 ASCII 및 바이너리 모두를 지원합니다. 바이너리 프로토콜을 사용하여 자동 검색 지원을 추가하려면 [ElastiCache 클러스터 클라이언트에 대한 소스 코드](#)를 참조하십시오.

구문

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

옵션

이름	설명	필수
sub-command	캐시 노드와 상호 작용하는 데 사용되는 하위 명령입니다. Auto Discovery의 경우 이 하위 명령은 <code>get</code> 입니다.	예
key	클러스터 구성이 저장되어 있는 키입니다. Auto Discovery의 경우 이 키 이름은 <code>cluster</code> 입니다.	예

클러스터 구성 정보를 보려면 다음 명령을 사용합니다.

```
config get cluster
```

캐시 엔진 버전 1.4.14 미만

클러스터 구성 정보를 보려면 다음 명령을 사용합니다.

```
get AmazonElastiCache:cluster
```

Note

"AmazonElastiCache:cluster" 키에는 클러스터 구성 정보가 상주하므로 이 키를 함부로 변경하지 마십시오. 이 키를 덮어쓰면 ElastiCache가 자동으로 올바르게 구성 정보를 업데이트하기 전에 클라이언트가 짧은 시간 동안(15초 미만) 잘못 구성될 수 있습니다.

출력 형식

`config get cluster`를 사용하면, `get AmazonElastiCache:cluster`를 사용하면 응답은 두 줄로 구성됩니다.

- 구성 정보의 버전 번호. 매번 노드가 캐시 클러스터에서 추가 또는 제거될 때마다 버전 번호가 하나씩 증가합니다.
- 캐시 노드 목록. 목록의 각 노드는 `hostname|ip-address|port` 그룹으로 표현되며 각 노드는 공백으로 구분됩니다.

캐리지 리턴 및 줄 바꿈 문자(CR + LF)는 각 행의 끝에 표시됩니다. 데이터 행 끝에 줄 바꿈 문자(LF)가 포함되며 여기에 CR + LF가 추가됩니다. 구성 버전 행은 CR 없이 LF로 종결됩니다.

노드가 세 개인 캐시 클러스터는 다음과 같이 표현됩니다.

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

각 노드는 CNAME 및 사실 IP 주소 모두를 표시됩니다. CNAME은 항상 있으며 사실 IP 주소가 사용 불가능한 경우 표시되지 않지만 파이프 문자 "|"는 여전히 프린트됩니다.

Example

다음은 구성 정보를 쿼리할 때 반환되는 페이로드의 예입니다.

```
CONFIG cluster 0 147\r\n12\nmyCluster.pc4ldq.0001.use1.cache.amazonaws.com|10.82.235.120|11211\nmyCluster.pc4ldq.0002.use1.cache.amazonaws.com|10.80.249.27|11211\n\r\nEND\r\n
```

Note

- 두 번째 행은 구성 정보가 이제까지 12회 수정되었음을 나타냅니다.
- 세 번째 행에서는 노드 목록이 호스트 이름의 사전순으로 지정됩니다. 이 정렬 순서는 클라이언트 애플리케이션에서 현재 사용 중인 것과 다른 순서로 되어 있을 수도 있습니다.

ElastiCache 자동 검색이 있는 클라이언트

이 단원에서는 ElastiCache PHP 및 .NET 클라이언트의 설치 및 구성에 대해 설명합니다.

주제

- [클러스터 클라이언트 설치 및 컴파일 \(p. 56\)](#)
- [ElastiCache 클라이언트 구성 \(p. 66\)](#)

클러스터 클라이언트 설치 및 컴파일

이 섹션에서는 PHP 및 .NET Amazon ElastiCache Auto Discovery 클러스터 클라이언트의 설치, 구성 및 컴파일을 다룹니다.

주제

- [.NET용 ElastiCache 클러스터 클라이언트 설치 \(p. 56\)](#)
- [PHP용 ElastiCache 클러스터 클라이언트 설치 \(p. 58\)](#)
- [PHP용 ElastiCache 클러스터 클라이언트에 대한 소스 코드 컴파일 \(p. 64\)](#)

.NET용 ElastiCache 클러스터 클라이언트 설치

<https://github.com/aws-labs/elasticache-cluster-config-net>에서 ElastiCache .NET 클러스터 클라이언트 코드를 오픈 소스로 찾을 수 있습니다.

이 단원에서는 Amazon EC2 인스턴스에서 ElastiCache 클러스터 클라이언트에 대한 .NET 구성 요소를 설치, 업데이트 및 제거하는 방법에 대해 설명합니다. Auto Discovery에 대한 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오. 클라이언트를 사용할 샘플 .NET 코드는 [.NET용 ElastiCache 클러스터 클라이언트 사용 \(p. 53\)](#) 단원을 참조하십시오.

주제

- [.NET 설치 \(p. 57\)](#)
- [ElastiCache용 ElastiCache .NET 클러스터 클라이언트 다운로드 \(p. 57\)](#)
- [NuGet으로 AWS 어셈블리 설치 \(p. 57\)](#)

.NET 설치

ElastiCache용 AWS .NET SDK를 사용하려면 .NET 3.5 이상이 설치되어 있어야 합니다. .NET 3.5 이상이 설치되어 있지 않은 경우 <http://www.microsoft.com/net>에서 최신 버전을 다운로드하여 설치할 수 있습니다.

ElastiCache용 ElastiCache .NET 클러스터 클라이언트 다운로드

ElastiCache .NET 클러스터 클라이언트를 다운로드하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 [ElastiCache Cluster Client]를 클릭합니다.
3. [Download ElastiCache Memcached Cluster Client] 목록에서 [.NET]을 선택한 다음 [Download]를 클릭합니다.

NuGet으로 AWS 어셈블리 설치

NuGet은 .NET 플랫폼을 위한 패키지 관리 시스템입니다. NuGet은 어셈블리 종속성을 인식하며 모든 필수 파일을 자동으로 설치합니다. NuGet이 설치된 어셈블리는 Program Files와 같은 중앙 위치가 아닌 솔루션에 저장되므로 호환성 문제 없이 애플리케이션에 특정한 버전을 설치할 수 있습니다.

NuGet 설치

MSDN의 Installation Gallery(<https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c> 참조)에서 NuGet을 설치할 수 있습니다. Visual Studio 2010 이상을 사용하는 경우 NuGet이 자동으로 설치됩니다.

[Solution Explorer] 또는 [Package Manager Console]에서 NuGet을 사용할 수 있습니다.

Solution Explorer에서 NuGet 사용

Visual Studio 2010의 Solution Explorer에서 NuGet을 사용하려면

1. [Tools] 메뉴에서 [Library Package Manager]를 선택합니다.
2. [Package Manager Console]을 클릭합니다.

Visual Studio 2012 또는 Visual Studio 2013의 Solution Explorer에서 NuGet을 사용하려면

1. [Tools] 메뉴에서 [NuGet Package Manager]를 선택합니다.
2. [Package Manager Console]을 클릭합니다.

명령줄에서 다음에 표시된 대로 Install-Package를 사용하여 어셈블리를 설치할 수 있습니다.

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

AWSSDK 및 AWS.Extensions 어셈블리 등 NuGet을 통해 사용할 수 있는 모든 패키지에 대한 페이지를 보려면 <http://www.nuget.org>의 NuGet 웹 사이트를 참조하십시오. 각 패키지의 페이지에는 콘솔을 사용하여 패키

지를 설치하기 위한 샘플 명령줄과 NuGet을 통해 사용할 수 있는 패키지의 이전 버전 목록이 포함되어 있습니다.

[Package Manager Console] 명령에 대한 자세한 내용은 <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>를 참조하십시오.

PHP용 ElastiCache 클러스터 클라이언트 설치

이 단원에서는 Amazon EC2 인스턴스에서 ElastiCache 클러스터 클라이언트에 대한 PHP 구성 요소를 설치, 업데이트 및 제거하는 방법에 대해 설명합니다. Auto Discovery에 대한 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#)를 참조하십시오. 클라이언트를 사용할 샘플 PHP 코드는 [PHP용 ElastiCache 클러스터 클라이언트 사용 \(p. 52\)](#) 단원을 참조하십시오.

주제

- [설치 패키지 다운로드 \(p. 58\)](#)
- [이미 php-memcached 확장 프로그램을 설치한 사용자의 경우 \(p. 59\)](#)
- [신규 사용자를 위한 설치 단계 \(p. 59\)](#)
- [PHP 클러스터 클라이언트 삭제 \(p. 64\)](#)

설치 패키지 다운로드

올바른 버전의 PHP용 ElastiCache 클러스터 클라이언트를 사용하는지 확인하려면 Amazon EC2 인스턴스에 설치된 PHP 버전을 알아야 합니다. Amazon EC2 인스턴스에서 실행되는 Linux 버전이 64비트인지 32비트인지도 알아야 합니다.

Amazon EC2 인스턴스에 설치되어 있는 PHP 버전을 확인하려면

- 명령 프롬프트에서 다음 명령을 실행합니다.

```
php -v
```

PHP 버전은 다음 예와 같이 출력에 표시됩니다.

```
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

PHP 및 Memcached 버전이 호환되지 않는 경우 다음과 같은 오류 메시지가 표시됩니다.

```
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
```

이러한 경우 소스 코드에서 모듈을 컴파일해야 합니다. 자세한 내용은 [PHP용 ElastiCache 클러스터 클라이언트에 대한 소스 코드 컴파일 \(p. 64\)](#) 단원을 참조하십시오.

Amazon EC2 AMI 아키텍처(64비트 또는 32비트)를 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 인스턴스 목록에서 Amazon EC2 인스턴스를 클릭합니다.

3. Description 탭에서 AMI: 필드를 찾습니다. 64비트 인스턴스는 설명에 x86_64를 포함해야 하며 32비트 인스턴스는 이 필드에 i386 또는 i686를 포함합니다.

이제 ElastiCache 클러스터 클라이언트를 다운로드할 준비가 되었습니다.

PHP용 ElastiCache 클러스터 클라이언트를 다운로드하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. ElastiCache 콘솔에서 ElastiCacheCluster Client(클러스터 클라이언트)를 선택합니다.
3. ElastiCache Memcached 클러스터 클라이언트 다운로드 목록에서 PHP 버전 및 AMI 아키텍처에 맞는 ElastiCache 클러스터 클라이언트를 선택한 다음 다운로드 버튼을 선택합니다.

이미 php-memcached 확장 프로그램을 설치한 사용자의 경우

php-memcached 설치를 업데이트하려면

1. [PHP 클러스터 클라이언트 삭제 \(p. 64\)](#) 항목에 명시된 대로 이전에 설치된 PHP용 Memcached 확장명을 제거합니다.
2. 이전에 [신규 사용자를 위한 설치 단계 \(p. 59\)](#)에 명시된 대로 새 ElastiCache php-memcached 확장명을 설치합니다.

신규 사용자를 위한 설치 단계

주제

- [신규 사용자를 위한 PHP 7.x 설치 \(p. 59\)](#)
- [신규 사용자를 위한 PHP 5.x 설치 \(p. 61\)](#)

신규 사용자를 위한 PHP 7.x 설치

주제

- [Ubuntu Server 14.04 LTS AMI\(64비트 및 32비트\)에서 PHP 7을 설치하려면 \(p. 59\)](#)
- [Amazon Linux 201609 AMI에서 PHP 7을 설치하려면 \(p. 60\)](#)
- [SUSE Linux AMI에서 PHP 7을 설치하려면 \(p. 61\)](#)

Ubuntu Server 14.04 LTS AMI(64비트 및 32비트)에서 PHP 7을 설치하려면

1. AMI에서 새 인스턴스를 시작합니다.
2. 다음 명령을 실행합니다:

```
sudo apt-get update
sudo apt-get install gcc g++
```

3. PHP 7을 설치합니다.

```
sudo yum install php70
```

4. Amazon ElastiCache 클러스터 클라이언트를 다운로드합니다.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. latest-64bit를 추출합니다.

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
클러스터 생성

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

6. 루트 권한으로 추출된 아티팩트 파일 `amazon-elasticache-cluster-client.so`를 `/usr/lib/php/20151012`로 복사합니다.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib/php/20151012
```

7. `extension=amazon-elasticache-cluster-client.so` 라인을 `/etc/php/7.0/cli/php.ini` 파일에 삽입합니다.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php/7.0/cli/php.ini
```

8. Apache 서버를 시작하거나 다시 시작합니다.

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

Amazon Linux 201609 AMI에서 PHP 7을 설치하려면

1. AMI에서 새 인스턴스를 시작합니다.
2. 다음 명령을 실행합니다.

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

3. PHP 7을 설치합니다.

```
sudo yum install php70
```

4. Amazon ElastiCache 클러스터 클라이언트를 다운로드합니다.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. `latest-64bit`를 추출합니다.

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

6. 루트 권한으로 추출된 아티팩트 파일 `amazon-elasticache-cluster-client.so`를 `/usr/lib64/php/7.0/modules/`로 복사합니다.

```
sudo mv artifact/amazon-elasticache-cluster-client.so /usr/lib64/php/7.0/modules/
```

7. `50-memcached.ini` 파일을 생성합니다.

```
echo "extension=amazon-elasticache-cluster-client.so" | sudo tee --append /etc/php-7.0.d/50-memcached.ini
```

8. Apache 서버를 시작하거나 다시 시작합니다.

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

SUSE Linux AMI에서 PHP 7을 설치하려면

1. AMI에서 새 인스턴스를 시작합니다.
2. 다음 명령을 실행합니다.

```
sudo zypper install gcc
```

3. PHP 7을 설치합니다.

```
sudo yum install php70
```

4. Amazon ElastiCache 클러스터 클라이언트를 다운로드합니다.

```
wget https://elasticache-downloads.s3.amazonaws.com/ClusterClient/PHP-7.0/latest-64bit
```

5. latest-64bit를 추출합니다.

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

6. 루트 권한으로 추출된 아티팩트 파일 amazon-elasticache-cluster-client.so를 /usr/lib64/php7/extensions/로 복사합니다.

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

7. extension=amazon-elasticache-cluster-client.so 라인을 /etc/php7/cli/php.ini 파일에 삽입합니다.

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

8. Apache 서버를 시작하거나 다시 시작합니다.

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

신규 사용자를 위한 PHP 5.x 설치

주제

- [Amazon Linux AMI 2014.03\(64비트 및 32비트\)에서 PHP 5를 설치하려면](#) (p. 61)
- [Red Hat Enterprise Linux 7.0 AMI\(64비트 및 32비트\)에서 PHP 5를 설치하려면](#) (p. 62)
- [Ubuntu Server 14.04 LTS AMI\(64비트 및 32비트\)에서 PHP 5를 설치하려면](#) (p. 62)
- [SUSE Linux Enterprise Server 11 AMI\(64비트 또는 32비트\)용 PHP 5를 설치하려면](#) (p. 63)
- [기타 Linux 배포](#) (p. 64)

Amazon Linux AMI 2014.03(64비트 및 32비트)에서 PHP 5를 설치하려면

1. Amazon Linux 인스턴스(64비트 또는 32비트)를 시작하여 인스턴스에 로그인합니다.
2. PHP 종속 파일을 설치합니다.

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

3. Amazon EC2 인스턴스 및 PHP 버전용 올바른 php-memcached 패키지를 다운로드합니다. 자세한 내용은 [설치 패키지 다운로드](#) (p. 58) 단원을 참조하십시오.

4. php-memcached를 설치합니다. URI는 설치 패키지에 대한 다운로드 경로여야 합니다.

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

다음은 PHP 5.4, 64비트 Linux용 샘플 설치 명령입니다. 이 샘플에 사용된 *X.Y.Z*를 실제 버전 번호로 교체해야 합니다.

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

Note

최신 버전의 설치 아티팩트를 사용하십시오.

5. 루트/sudo 권한으로 /etc/php.d 디렉터리에 이름이 memcached.ini인 새 파일을 추가하고 파일에 "extension=amazon-elasticache-cluster-client.so"를 삽입합니다.

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

6. Apache 서버를 시작하거나 다시 시작합니다.

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

Red Hat Enterprise Linux 7.0 AMI(64비트 및 32비트)에서 PHP 5를 설치하려면

1. Red Hat Enterprise Linux 인스턴스(64비트 또는 32비트)를 시작하여 인스턴스에 로그인합니다.
2. PHP 종속 파일을 설치합니다.

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

3. Amazon EC2 인스턴스 및 PHP 버전용 올바른 php-memcached 패키지를 다운로드합니다. 자세한 내용은 [설치 패키지 다운로드 \(p. 58\)](#) 단원을 참조하십시오.
4. php-memcached를 설치합니다. URI는 설치 패키지에 대한 다운로드 경로여야 합니다.

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

5. 루트/sudo 권한으로 /etc/php.d 디렉터리에 이름이 memcached.ini인 새 파일을 추가하고 파일에 extension=amazon-elasticache-cluster-client.so를 삽입합니다.

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

6. Apache 서버를 시작하거나 다시 시작합니다.

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

Ubuntu Server 14.04 LTS AMI(64비트 및 32비트)에서 PHP 5를 설치하려면

1. Ubuntu Linux 인스턴스(64비트 또는 32비트)를 시작하여 인스턴스에 로그인합니다.
2. PHP 종속 파일을 설치합니다.

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

3. Amazon EC2 인스턴스 및 PHP 버전용 올바른 php-memcached 패키지를 다운로드합니다. 자세한 내용은 [설치 패키지 다운로드 \(p. 58\)](#) 단원을 참조하십시오.
4. php-memcached를 설치합니다. URI는 설치 패키지에 대한 다운로드 경로여야 합니다.

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

Note

이 설치 단계에서는 빌드 아티팩트 `amazon-elasticache-cluster-client.so`를 `/usr/lib/php5/20121212*` 디렉터리에 설치합니다. 다음 단계에서 필요한 빌드 아티팩트의 절대 경로를 확인하십시오.

이전 명령이 작동하지 않으면 다운로드된 *.tgz 파일에서 PHP 클라이언트 아티팩트 `amazon-elasticache-cluster-client.so`를 수동으로 추출하여 `/usr/lib/php5/20121212*` 디렉터리에 복사해야 합니다.

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

5. 루트/sudo 권한으로 `/etc/php5/cli/conf.d` 디렉터리에 이름이 `memcached.ini`인 새 파일을 추가하고 파일에 `extension=<absolute path to amazon-elasticache-cluster-client.so>`를 삽입합니다.

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

6. Apache 서버를 시작하거나 다시 시작합니다.

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

SUSE Linux Enterprise Server 11 AMI(64비트 또는 32비트)용 PHP 5를 설치하려면

1. SUSE Linux 인스턴스(64비트 또는 32비트)를 시작하여 인스턴스에 로그인합니다.
2. PHP 종속 파일을 설치합니다.

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

3. Amazon EC2 인스턴스 및 PHP 버전용 올바른 php-memcached 패키지를 다운로드합니다. 자세한 내용은 [설치 패키지 다운로드 \(p. 58\)](#) 단원을 참조하십시오.
4. php-memcached를 설치합니다. URI는 설치 패키지에 대한 다운로드 경로여야 합니다.

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

5. 루트/sudo 권한으로 `/etc/php5/conf.d` 디렉터리에 이름이 `memcached.ini`인 새 파일을 추가하고 파일에 `extension=amazon-elasticache-cluster-client.so`를 삽입합니다.

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

6. Apache 서버를 시작하거나 다시 시작합니다.

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

Note

이전의 모든 플랫폼에 대해 5단계가 작동하지 않으면 `amazon-elasticache-cluster-client.so`의 설치 경로를 확인하고 확장명에 바이너리의 전체 경로를 지정하십시오. 또한 사용 중인 PHP가 지원되는 버전인지 확인하십시오. 5.3 ~ 5.5의 버전이 지원됩니다.

기타 Linux 배포

일부 시스템(특히 CentOS 및 Red Hat Enterprise Linux (RHEL) 7.1)에서는 `libsas12.so.3`이 `libsas12.so.2`로 대체되었습니다. 이러한 시스템에서 ElastiCache 클러스터 클라이언트를 로드하면 `libsas12.so.2`를 찾아 로드하려고 시도하지만 실패합니다. 이 문제를 해결하려면 `libsas12.so.3`에 대한 심볼 링크를 생성합니다. 그러면 클라이언트가 `libsas12.so.2`를 로드하려고 시도할 때 `libsas12.so.3`으로 리디렉션됩니다. 다음 코드는 이 심볼 링크를 생성합니다.

```
cd /usr/lib64
$ sudo ln libsas12.so.3 libsas12.so.2
```

PHP 클러스터 클라이언트 삭제

주제

- [이전 버전의 PHP 7 제거 \(p. 64\)](#)
- [이전 버전의 PHP 5 제거 \(p. 64\)](#)

이전 버전의 PHP 7 제거

이전 버전의 PHP 7을 제거하려면

1. 이전에 설치 지침에 표시된 대로 해당하는 PHP 라이브러리 디렉터리에서 `amazon-elasticache-cluster-client.so` 파일을 제거합니다. [이미 php-memcached 확장 프로그램을 설치한 사용자의 경우 \(p. 59\)](#)의 설치에 대한 단원을 참조하십시오.
2. `php.ini` 파일에서 `extension=amazon-elasticache-cluster-client.so` 라인을 제거합니다.
3. Apache 서버를 시작하거나 다시 시작합니다.

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

이전 버전의 PHP 5 제거

이전 버전의 PHP 5를 제거하려면

1. `php-memcached` 확장 프로그램을 삭제합니다.

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

2. 이전 설치 단계에 표시된 대로 해당하는 디렉터리에 추가된 `memcached.ini` 파일을 제거합니다.

PHP용 ElastiCache 클러스터 클라이언트에 대한 소스 코드 컴파일

이 섹션에서는 PHP용 ElastiCache 클러스터 클라이언트에 대한 소스 코드를 획득 및 컴파일하는 방법을 다룹니다.

[aws-elasticache-cluster-client-libmemcached](#) 및 [aws-elasticache-cluster-client-memcached-for-php](#)의 두 패키지는 GitHub에서 가져와 컴파일해야 합니다.

주제

- [libmemcached 라이브러리 컴파일 \(p. 65\)](#)
- [PHP용 ElastiCache Memcached Auto Discovery Client 컴파일 \(p. 65\)](#)

libmemcached 라이브러리 컴파일

aws-elasticache-cluster-client-libmemcached 라이브러리를 컴파일하려면 다음을 수행하십시오.

1. Amazon EC2 인스턴스를 시작합니다.
2. 라이브러리 종속 항목을 설치합니다.

- Amazon Linux 201509 AMI의 경우

```
sudo yum install gcc gcc-c++ autoconf libevent-devel
```

- Ubuntu 14.04 AMI의 경우

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

3. 리포지토리를 가져오고 코드를 컴파일합니다.

```
Download and expand https://github.com/aws-labs/aws-elasticache-cluster-client-libmemcached/archive/v1.0.18.tar.gz  
cd aws-elasticache-cluster-client-libmemcached  
mkdir BUILD  
cd BUILD  
../configure --prefix=<libmemcached-install-directory> --with-pic  
make  
sudo make install
```

PHP용 ElastiCache Memcached Auto Discovery Client 컴파일

다음 섹션에서는 ElastiCache Memcached Auto Discovery Client를 컴파일하는 방법을 설명합니다.

주제

- [PHP 7용 ElastiCache Memcached Client 컴파일 \(p. 65\)](#)
- [PHP 5용 ElastiCache Memcached Client 컴파일 \(p. 66\)](#)

PHP 7용 ElastiCache Memcached Client 컴파일

코드 디렉터리에서 다음 명령 세트를 실행합니다.

```
git clone https://github.com/aws-labs/aws-elasticache-cluster-client-memcached-for-php.git  
cd aws-elasticache-cluster-client-memcached-for-php  
git checkout php7  
sudo yum install php70-devel  
phpize  
../configure --with-libmemcached-dir=<libmemcached-install-directory> --disable-memcached-sasl  
make  
make install
```

Note

다양한 Linux 플랫폼으로 이식될 수 있도록 libmemcached 라이브러리를 PHP 바이너리에 정적으로 연결할 수 있습니다. 이렇게 하려면 make 전에 다음 명령을 실행하십시오.

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

PHP 5용 ElastiCache Memcached Client 컴파일

aws-elasticache-cluster-client-memcached-for-php/ 폴더에서 다음 명령을 실행하여 aws-elasticache-cluster-client-memcached-for-php를 컴파일합니다.

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

ElastiCache 클라이언트 구성

ElastiCache 클러스터는 Memcached와 프로토콜이 호환됩니다. 현재 기존 Memcached 환경과 함께 사용하고 있는 코드, 애플리케이션 및 인기 도구는 서비스와 원활하게 연동합니다.

이 단원은 ElastiCache에 있는 캐시 노드에 연결하기 위한 특정 고려 사항을 설명합니다.

주제

- [노드 엔드포인트 및 포트 번호 찾기 \(p. 66\)](#)
- [Auto Discovery 사용을 위한 연결 \(p. 67\)](#)
- [DNS 이름 및 기본 IP \(p. 67\)](#)

노드 엔드포인트 및 포트 번호 찾기

캐시 노드에 연결하려는 애플리케이션은 해당 노드의 Endpoint 및 포트 번호를 알아야 합니다.

노드 엔드포인트 및 포트 번호 찾기 (콘솔)

노드 엔드포인트 및 포트 번호를 확인하려면

1. [Amazon ElastiCache Management Console](#)에 로그인한 후 클러스터에서 실행 중인 엔진을 선택합니다.
선택한 엔진을 실행하고 있는 모든 클러스터의 목록이 표시됩니다.
2. 실행 중인 엔진 및 구성에 대해 다음을 계속합니다.
3. 관심 있는 클러스터의 이름을 선택합니다.
4. 관심이 있는 노드에 대한 [Port] 및 [Endpoint] 열을 찾습니다.

캐시 노드 엔드포인트 및 포트 번호 찾기 (AWS CLI)

캐시 노드 엔드포인트 및 포트 번호를 확인하려면 describe-cache-clusters 명령을 --show-cache-node-info 파라미터와 함께 사용하십시오.

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

정규화된 DNS 이름 및 포트 번호는 출력의 엔드포인트 섹션에 있습니다.

캐시 노드 엔드포인트 및 포트 번호 찾기 (ElastiCache API)

캐시 노드 엔드포인트 및 포트 번호를 확인하려면 DescribeCacheClusters 작업을 ShowCacheNodeInfo=true 파라미터와 함께 사용하십시오.

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=AWS4-HMAC-SHA256  
&X-Amz-Credential=<credential>  
&X-Amz-Date=20140421T220302Z  
&X-Amz-Expires=20140421T220302Z  
&X-Amz-Signature=<signature>  
&X-Amz-SignedHeaders=Host
```

Auto Discovery 사용을 위한 연결

애플리케이션이 Auto Discovery를 사용하는 경우에는 각 캐시 노드에 대한 개별 Endpoint가 아니라 클러스터에 대한 구성 Endpoint만 알면 됩니다. 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오.

Note

이때, Auto Discovery는 Memcached를 실행하는 캐시 클러스터에 대해서만 사용할 수 있습니다.

DNS 이름 및 기본 IP

클라이언트는 캐시 데이터를 보관 중인 서버의 주소 및 포트 번호를 포함하는 서버 목록을 유지 관리합니다. ElastiCache를 사용할 때 DescribeCacheClusters API(또는 describe-cache-clusters 명령줄 유틸리티)는 서버 목록에 사용할 수 있는 정규화된 DNS 항목과 포트 번호를 반환합니다.

Important

중요한 것은 클라이언트 애플리케이션이 캐시 노드 Endpoint에 연결을 시도할 때 캐시 노드의 DNS 이름을 자주 확인하도록 구성되어 있다는 점입니다.

VPC 설치

ElastiCache는 실패 시 캐시 노드가 복구될 때 캐시 노드의 DNS 이름과 IP 주소가 모두 동일하게 유지되도록 합니다.

비 VPC 설치

ElastiCache는 실패 시 캐시 노드가 복구될 때 캐시 노드의 DNS 이름이 변경되지 않도록 하지만, 캐시 노드의 기본 IP 주소는 변경될 수 있습니다.

클라이언트 라이브러리는 대부분 기본적으로 영구 캐시 노드 연결을 지원합니다. ElastiCache를 사용할 때 할 때 영구 캐시 노드 연결을 사용하는 것이 좋습니다. 클라이언트 측 DNS 캐싱은 클라이언트 라이브러리, 언어 런타임 또는 클라이언트 운영 체제 등 여러 위치에서 발생할 수 있습니다. 캐시 노드의 IP 주소를 자주 확인하도록 하려면 각 계층의 애플리케이션 구성을 검토해야 합니다.

노드 크기 선택

클러스터에 대해 선택하는 노드 크기는 비용, 성능, 내결함성에 영향을 미칩니다.

Memcached 노드 크기 선택

Memcached 클러스터에는 여러 노드로 분할된 클러스터 데이터와 한 개 이상의 노드가 포함되어 있습니다. 클러스터의 메모리 요구와 노드의 메모리가 서로 관련은 있지만 동일하지는 않습니다. 큰 노드 두세 개나 작은 노드 여러 개를 두어 필요한 클러스터 메모리 용량을 확보할 수 있습니다. 또한 요구량이 달라지면 클러스터에서 노드를 추가하거나 제거하여 필요한 만큼만 비용을 지불할 수 있습니다.

클러스터의 총 메모리 용량은 시스템 오버헤드를 뺀 각 노드의 RAM 용량을 클러스터에 있는 노드 수에 곱하는 방식으로 계산됩니다. 각 노드의 용량은 노드 유형에 따라 다릅니다.

$$\text{cluster_capacity} = \text{number_of_nodes} * (\text{node_capacity} - \text{system_overhead})$$

클러스터의 노드 수는 Memcached를 실행하는 클러스터의 가용성을 결정하는 핵심 요인입니다. 단일 노드의 오류는 애플리케이션 가용성과 백엔드 데이터베이스에 대한 로드에도 영향을 미칠 수 있습니다. 이러한 경우, ElastiCache가 오류 캐시 노드를 대체하기 위해 프로비저닝하고 다시 채워집니다. 이러한 가용성 영향을 줄이려면 적은 수의 대용량 노드를 사용하는 대신 더 적은 용량의 더 여러 개의 노드로 메모리 및 컴퓨팅 용량을 분산해야 합니다.

캐시 메모리가 35GB인 시나리오에서 다음과 같은 구성으로 설정할 수 있습니다.

- 메모리 3.22GB인 `cache.t2.medium` 노드 11개에 스레드가 각각 2개이면 35.42GB, 스레드 22개와 같습니다.
- 메모리 6.42GB인 `cache.m4.large` 노드 6개에 스레드가 각각 2개이면 38.52GB, 스레드 12개와 같습니다.
- 메모리 12.3GB인 `cache.r4.large` 노드 3개에 스레드가 각각 2개이면 36.90GB, 스레드 6개와 같습니다.
- 메모리 14.28GB인 `cache.m4.xlarge` 노드 3개에 스레드가 각각 4개이면 42.84GB, 스레드 12개와 같습니다.

노드 옵션 비교

노드 유형	메모리	코어	시간당 비용 *	필요한 노드	메모리 합계	총 코어	월별 비용
cache.t2.medium	3.22GB	2	0.068 USD	11	35.42GB	22	538.56 USD
cache.m4.large	6.42GB	2	0.156 USD	6	38.52GB	12	673.92 USD
cache.m4.xlarge	14.28GB	4	0.311 USD	3	42.84GB	12	671.76 USD
cache.m5.xlarge	12.93GB	4	0.311 USD	3	38.81GB	12	671.76 USD
cache.r4.large	12.3GB	2	0.228 USD	3	36.9GB	6	492.48 USD
cache.r5.large	13.07GB	2	0.216 USD	3	39.22GB	6	466.56 USD

* 2018년 10월 22일 기준 각 노드의 시간당 비용

노드 유형	메모리	코어	시간당 비용 *	필요한 노드	메모리 합계	총 코어	월별 비용
30일간(720시간) 사용량 100%의 월별 비용							

여기에 나와 있는 옵션은 각각 비슷한 메모리 용량을 제공하지만 컴퓨팅 용량과 비용은 다릅니다. 옵션별 비용을 비교하려면 [Amazon ElastiCache 요금](#)을 참조하십시오.

Memcached를 실행하는 클러스터의 경우 각 노드의 사용 가능한 메모리 일부가 연결 오버헤드에 사용됩니다. 자세한 내용은 [Memcached 연결 오버헤드 \(p. 136\)](#) 단원을 참조하십시오.

여러 노드를 사용하면 노드에 키를 분산시켜야 합니다. 노드마다 고유한 엔드포인트가 있습니다. ElastiCache 자동 검색 기능을 사용하면 클라이언트 프로그램이 클러스터의 모든 노드를 자동으로 식별하여 엔드포인트를 쉽게 관리할 수 있습니다. 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오.

경우에 따라 얼마나 많은 용량이 필요한지 확신할 수 없을 수도 있습니다. 그렇다면, 테스트를 위해 1개의 cache.m5.large 노드를 사용하는 것이 좋습니다. 그런 다음 Amazon CloudWatch에 게시된 ElastiCache 측정치를 사용하여 메모리 사용량, CPU 사용률 및 캐시 적중률을 모니터링합니다. ElastiCache의 CloudWatch 지표에 대한 자세한 내용은 [측정치를 통한 모니터링 사용 \(p. 139\)](#) 단원을 참조하십시오. 프로덕션 및 대규모 워크로드에는 R5 노드의 성능과 RAM 비용 가치가 가장 우수합니다.

클러스터의 적중률이 기대에 미치지 못하면 간편하게 노드를 더 추가하여 클러스터의 총 가용 메모리를 늘릴 수 있습니다.

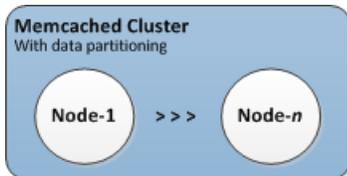
클러스터가 CPU의 제약을 받지만 적중률은 충분할 경우 컴퓨팅 파워가 더 큰 노드 유형으로 새로운 클러스터를 설정해 보십시오.

Memcached 클러스터 생성 (콘솔)

Memcached 엔진을 사용하면 Amazon ElastiCache가 여러 노드에 데이터를 가로로 분할하도록 지원합니다. Memcached는 자동 검색을 지원하므로 각 노드의 엔드포인트를 추적할 필요가 없습니다. Memcached가 각 노드의 엔드포인트를 추적하여 노드가 추가되거나 제거되면 엔드포인트 목록을 업데이트합니다. 클러스터와 상호 작용이 필요한 모든 애플리케이션은 구성 엔드포인트입니다. 자동 검색에 대한 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오.

ElastiCache 콘솔을 사용하여 Memcached 클러스터를 생성하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에 있는 Amazon ElastiCache 콘솔을 여십시오.
2. 상단 오른쪽 모서리의 목록에서 이 클러스터를 시작할 AWS 리전을 선택합니다.
3. 탐색 창에서 [Memcached]를 선택합니다.
4. [Create]를 선택합니다.
5. [Cluster engine]에 대해 [Memcached]를 선택합니다. Memcached를 선택하면 다음과 같은 Memcached 클러스터가 생성됩니다. 5.f단계에서 선택한 노드 수에 따라 노드 수가 결정됩니다(최대 20개).



데이터 분할이 지원되는 Memcached 클러스터

6. [Memcached settings] 단원을 완료합니다.

- a. [Name]에 클러스터 이름을 입력합니다.

클러스터 명명 제약 조건은 다음과 같습니다.

- 1-40자의 영숫자 또는 하이픈으로 구성되어야 합니다.
- 문자로 시작해야 합니다.
- 하이픈 2개가 연속될 수 없습니다.
- 끝에 하이픈이 올 수 없습니다.

- b. [Engine version compatibility]에 대해 이 클러스터를 실행할 Memcached 엔진 버전을 선택합니다. 이전 버전을 실행할 특별한 이유가 없는 한 최신 버전을 선택하는 것이 좋습니다.

Important

새로운 엔진 버전으로 업그레이드할 수 있습니다. 자세한 내용은 [엔진 버전 업그레이드 \(p. 29\)](#) 단원을 참조하십시오. Memcached 엔진 버전 변경은 클러스터 데이터가 손실되는 번거로운 프로세스입니다.

- c. [Port]에서 기본 포트 11211을 허용합니다. 다른 포트를 사용하려면 포트 번호를 입력합니다.

- d. [Parameter group]에 대해 기본 파라미터 그룹을 선택하거나 이 클러스터에 사용할 파라미터 그룹을 선택하거나 [Create new]를 선택하여 이 클러스터에 사용할 새 파라미터 그룹을 생성합니다.

파라미터 그룹은 클러스터의 런타임 파라미터를 제어합니다. 파라미터 그룹에 대한 자세한 내용은 [Memcached 특정 파라미터 \(p. 127\)](#) 및 [파라미터 그룹 생성 \(p. 116\)](#) 단원을 참조하십시오.

- e. 노드 유형에서 아래쪽 화살표(▼)를 클릭합니다. [Change node type] 대화 상자에서 원하는 노드 유형의 인스턴스 패밀리를 선택하고 이 클러스터에 사용할 노드 유형을 선택한 다음 [Save]를 선택합니다.

자세한 내용은 [Memcached 노드 크기 선택 \(p. 68\)](#) 단원을 참조하십시오.

- f. [Number of nodes]에 대해 이 클러스터에 사용할 노드 수를 선택합니다. 클러스터 노드에 데이터를 분할합니다.

나중에 노드 수를 변경하려면 Memcached를 사용하여 쉽게 가로 조정을 수행할 수 있습니다. 자세한 내용은 [ElastiCache for Memcached 클러스터 조정 \(p. 113\)](#) 단원을 참조하십시오.

7. [Advanced Memcached settings]를 클릭하고 단원을 완료합니다.

- a. [Subnet group]에 대해 이 클러스터에 적용할 서브넷을 선택합니다.

자세한 내용은 [서브넷 및 서브넷 그룹 \(p. 182\)](#) 단원을 참조하십시오.

- b. [Availability zone(s)]에 대해 다음 두 가지 옵션이 있습니다.

- 기본 설정 없음 - ElastiCache에서 클러스터에 있는 각 노드의 가용 영역을 선택합니다.
- Specify availability zones(가용 영역 지정) - 클러스터에 있는 각 노드의 가용 영역을 지정합니다.

각 노드의 가용 영역을 지정하기로 선택한 경우 각 노드 이름 오른쪽에 있는 목록에서 가용 영역을 선택하십시오.

여러 가용 영역에서 노드를 찾으면 내결함성이 개선됩니다. 자세한 내용은 [가용 영역 장애 완화 \(p. 213\)](#) 단원을 참조하십시오.

자세한 내용은 [리전 및 가용 영역 선택 \(p. 30\)](#) 단원을 참조하십시오.

- c. [Security groups]에 대해 이 클러스터에 적용할 보안 그룹을 선택합니다.

자세한 내용은 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

- d. 유지 관리 기간은 ElastiCache가 클러스터의 시스템 유지 관리를 예약하는 시간이며 일반적으로 매주 한 시간입니다. ElastiCache에서 유지 관리 기간의 요일과 시간을 선택하도록 허용하거나(기

본 설정 없음) 요일, 시간 및 기간을 직접 선택할 수 있습니다(Specify maintenance window(유지 관리 기간 지정)). 유지 관리 기간 지정을 선택한 경우 유지 관리 기간의 [Start day], [Start time] 및 [Duration](시간)을 선택합니다. 모든 시간은 UCT 시간입니다.

자세한 내용은 [유지 관리 관리 중 \(p. 33\)](#) 단원을 참조하십시오.

- e. 알림에 대해 기존의 Amazon Simple Notification Service (Amazon SNS) 항목을 선택하거나 수동 ARN 입력을 선택하고 Amazon 리소스 이름(ARN) 항목을 입력합니다. Amazon SNS를 통해 인터넷에 연결된 스마트 디바이스에 알림을 푸시할 수 있습니다. 기본적으로 알림이 비활성화됩니다. 자세한 내용은 <https://aws.amazon.com/sns/>를 참조하십시오.
8. 입력 및 선택한 내용을 모두 검토한 다음 돌아가 필요한 내용을 수정합니다. 준비가 되면 [Create]를 선택해 클러스터를 시작합니다.

클러스터 상태가 가용이 되면 Amazon EC2 액세스 권한을 부여하고 클러스터에 연결하며 사용할 수 있습니다. 자세한 내용은 [2단계: 액세스 자격 부여 \(p. 16\)](#) 및 [3단계: 클러스터 노드에 연결 \(p. 16\)](#) 단원을 참조하십시오.

Important

클러스터를 사용할 수 있게 되면 클러스터를 적극 사용하지 않더라도 클러스터가 활성화되어 있는 매 시간 또는 60분 미만 단위로 비용이 청구됩니다. 이 클러스터의 요금 발생을 중지하려면 클러스터를 삭제해야 합니다. [클러스터 삭제하기 \(p. 88\)](#) 단원을 참조하십시오.

클러스터 생성 (AWS CLI)

AWS CLI를 사용하여 클러스터를 생성하려면 `create-cache-cluster` 명령을 사용하십시오.

Important

클러스터를 사용할 수 있게 되면 클러스터를 적극 사용하지 않더라도 클러스터가 활성화되어 있는 매 시간 또는 60분 미만 단위로 비용이 청구됩니다. 이 클러스터의 요금 발생을 중지하려면 클러스터를 삭제해야 합니다. [클러스터 삭제하기 \(p. 88\)](#) 단원을 참조하십시오.

주제

- [Memcached 캐시 클러스터 \(AWS CLI\) 생성 \(p. 71\)](#)

Memcached 캐시 클러스터 (AWS CLI) 생성

다음 CLI 코드를 통해 노드가 3개인 Memcached 캐시 클러스터가 생성됩니다.

Linux, macOS, or Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

클러스터 생성 (ElastiCache API)

ElastiCache API를 사용하여 클러스터를 생성하려면 `CreateCacheCluster` 작업을 사용하십시오.

Important

클러스터를 사용할 수 있게 되면 클러스터를 사용하지 않더라도 클러스터가 활성화되어 있는 매 시간 또는 60분 미만 단위로 비용이 청구됩니다. 이 클러스터의 요금 발생을 중지하려면 클러스터를 삭제해야 합니다. [클러스터 삭제하기 \(p. 88\)](#) 단원을 참조하십시오.

주제

- [Memcached 캐시 클러스터 \(ElastiCache API\) 생성 \(p. 72\)](#)

Memcached 캐시 클러스터 (ElastiCache API) 생성

다음 코드는 3개 노드가 있는 Memcached 클러스터를 생성합니다 (ElastiCache API).

읽기 쉽도록 줄바꿈이 추가됩니다.

```
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=AWS4-HMAC-SHA256  
&X-Amz-Credential=<credential>  
&X-Amz-Date=20150508T220302Z  
&X-Amz-Expires=20150508T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Signature=<signature>
```

클러스터 세부 정보 보기

ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 클러스터 하나 이상에 대한 세부 정보를 볼 수 있습니다.

주제

- [클러스터 세부 정보 보기 \(콘솔\) \(p. 72\)](#)
- [클러스터 세부 정보 보기 \(AWS CLI\) \(p. 73\)](#)
- [클러스터 세부 정보 보기 \(ElastiCache API\) \(p. 75\)](#)

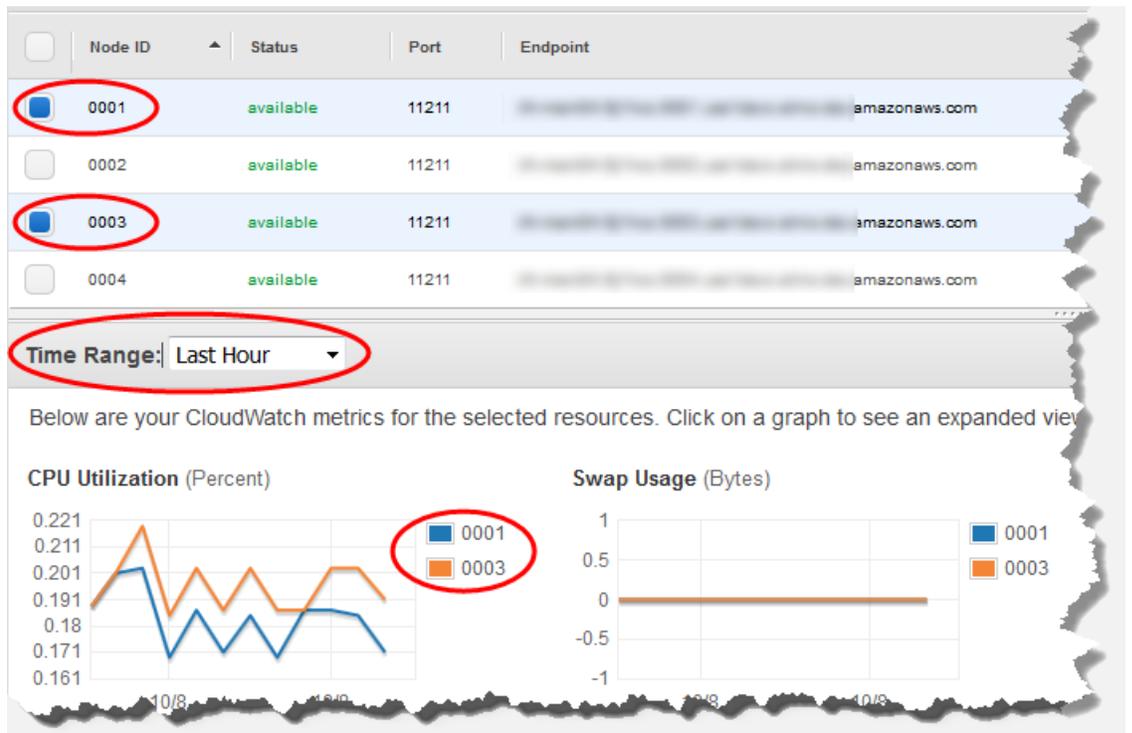
클러스터 세부 정보 보기 (콘솔)

ElastiCache 콘솔, ElastiCache용 AWS CLI 또는 ElastiCache API를 사용하여 Memcached 클러스터의 세부 정보를 볼 수 있습니다.

다음 절차에서는 ElastiCache 콘솔을 사용하여 Memcached 클러스터의 세부 정보를 보는 방법을 자세히 설명합니다.

Memcached 클러스터의 세부 정보를 보려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에 있는 Amazon ElastiCache 콘솔을 여십시오.
2. 상단 오른쪽 모서리의 목록에서 원하는 AWS 리전을 선택합니다.
3. ElastiCache 콘솔 대시보드에서 Memcached를 선택합니다. 그러면 버전에 관계없이 Memcached를 실행하는 모든 클러스터 목록이 표시됩니다.
4. 클러스터의 세부 정보를 보려면 클러스터 이름의 왼쪽에 있는 상자를 선택합니다.
5. 노드 정보를 보려면
 - a. 클러스터의 이름을 선택합니다.
 - b. [Nodes] 탭을 선택합니다.
 - c. 노드 1개 이상의 지표를 보려면 노드 ID 왼쪽의 상자를 선택한 후 [Time range] 목록에서 지표의 시간 범위를 선택합니다. 여러 노드를 선택하면 중첩 그래프가 생성됩니다.



Memcached 노드 2개에 대한 지난 1시간의 지표

클러스터 세부 정보 보기 (AWS CLI)

AWS CLI `describe-cache-clusters` 명령을 사용하여 클러스터의 세부 정보를 볼 수 있습니다. `--cache-cluster-id` 파라미터가 생략되면 여러 클러스터(최대 `--max-items`개)의 세부 정보가 반환됩니다. `--cache-cluster-id` 파라미터가 포함되면 지정한 클러스터의 세부 정보가 반환됩니다. `--max-items` 파라미터를 사용하여 반환되는 레코드 수를 제한할 수 있습니다.

다음 코드는 `my-cluster`의 세부 정보를 나열합니다.

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

다음 코드는 클러스터 최대 25개의 세부 정보를 나열합니다.

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
클러스터 세부 정보 보기

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

Example

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

이 작업은 다음과 비슷한 출력을 생성합니다(JSON 형식).

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

자세한 내용은 AWS CLI의 ElastiCache 항목 [describe-cache-clusters](#)를 참조하십시오.

클러스터 세부 정보 보기 (ElastiCache API)

ElastiCache API `DescribeCacheClusters` 작업을 사용하여 클러스터의 세부 정보를 볼 수 있습니다. `CacheClusterId` 파라미터가 포함되면 지정한 클러스터의 세부 정보가 반환됩니다. `CacheClusterId` 파라미터가 생략되면 클러스터 최대 `MaxRecords`개(기본값 100)의 세부 정보가 반환됩니다. `MaxRecords`의 값은 20 이상 또는 100 이하여야 합니다.

다음 코드는 `my-cluster`의 세부 정보를 나열합니다.

```
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>
```

다음 코드는 클러스터 최대 25개의 세부 정보를 나열합니다.

```
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>
```

자세한 내용은 ElastiCache API 참조 항목 [DescribeCacheClusters](#)을 참조하십시오.

ElastiCache 클러스터 수정

클러스터에서 노드를 추가하거나 제거하는 것 외에도 보안 그룹을 추가하거나 유지 관리 기간 또는 파라미터 그룹을 변경하는 등 기존의 클러스터를 변경해야 할 경우도 있습니다.

유지 관리 기간을 사용률이 가장 낮은 시간으로 낮추는 것이 유익하므로 수정해야 할 때도 있습니다.

클러스터의 파라미터를 변경하면 변경 사항은 또는 클러스터가 재시작된 즉시 또는 그 이후에 클러스터에 적용됩니다. 이는 클러스터의 파라미터 그룹 자체에서 변경하든 파라미터 값을 클러스터의 파라미터 그룹 내에서 변경하든 마찬가지입니다. 특정 파라미터의 변경 사항이 적용되는 시점을 확인하려면 [Memcached 특정 파라미터 \(p. 127\)](#) 및 에 대한 표에서 변경 사항 적용 열을 참조하십시오. 클러스터 재부팅에 관한 자세한 내용은 [클러스터 재부팅 \(p. 77\)](#) 단원을 참조하십시오.

AWS Management 콘솔 사용

클러스터를 삭제하려면 (console)

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 상단 오른쪽 모서리의 목록에서 수정하려는 클러스터가 있는 AWS 리전을 선택합니다.
3. 탐색 창에서 수정하려는 클러스터에서 실행 중인 엔진을 선택합니다.

선택한 엔진의 클러스터 목록이 나타납니다.

4. 클러스터 목록에서 수정하고자 하는 클러스터의 이름 왼쪽에 있는 상자가 아닌 클러스터의 이름을 선택합니다.
5. [Modify]를 선택합니다.

[Modify Cluster] 창이 나타납니다.

6. [Modify Cluster] 창에서 원하는 내용을 수정합니다.

Important

새로운 엔진 버전으로 업그레이드할 수 있습니다. 이에 대한 자세한 내용은 [엔진 버전 업그레이드 \(p. 29\)](#) 단원을 참조하십시오. 하지만 기존의 클러스터를 삭제하고 새로 만들지 않는 한 이전 엔진 버전으로 다운그레이드할 수 없습니다.

[Apply Immediately] 상자는 엔진 버전 수정에만 적용됩니다. 변경 사항을 즉시 적용하려면 Apply Immediately(즉시 적용) 확인란을 선택합니다. 이 상자를 선택하지 않으면 다음 번 유지 관리 기간에 엔진 버전 수정이 적용됩니다. 유지 관리 기간 변경과 같은 다른 수정은 즉시 적용됩니다.

7. [Modify]를 선택합니다.

AWS CLI 사용

AWS CLI `modify-cache-cluster` 작업을 사용하여 기존의 클러스터를 수정할 수 있습니다. 클러스터의 구성 값을 수정하려면 클러스터 ID, 변경할 파라미터 및 파라미터의 새 값을 지정합니다. 다음 예제에서는 `my-cluster`라는 클러스터의 유지 관리 기간을 변경하고 변경 사항을 즉시 적용합니다.

Important

새로운 엔진 버전으로 업그레이드할 수 있습니다. 이에 대한 자세한 내용은 [엔진 버전 업그레이드 \(p. 29\)](#) 단원을 참조하십시오. 하지만 기존의 클러스터를 삭제하고 새로 만들지 않는 한 이전 엔진 버전으로 다운그레이드할 수 없습니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache modify-cache-cluster \
```

```
--cache-cluster-id my-cluster \  
--preferred-maintenance-window sun:23:00-mon:02:00
```

Windows의 경우:

```
aws elasticache modify-cache-cluster ^  
--cache-cluster-id my-cluster ^  
--preferred-maintenance-window sun:23:00-mon:02:00
```

--apply-immediately 파라미터는 노드 유형, 엔진 버전의 수정 및 클러스터의 노드 수 변경에만 적용됩니다. 이 변경 사항을 즉시 적용하려면 --apply-immediately 파라미터를 사용하십시오. 다음 번 유지 관리 기간으로 이 변경을 연기하려면 --no-apply-immediately 파라미터를 사용하십시오. 유지 관리 기간 변경과 같은 다른 수정은 즉시 적용됩니다.

자세한 내용은 AWS CLI의 ElastiCache 항목 [modify-cache-cluster](#)를 참조하십시오.

ElastiCache API 사용

ElastiCache API ModifyCacheCluster 작업을 사용하여 기존의 클러스터를 수정할 수 있습니다. 클러스터의 구성 값을 수정하려면 클러스터 ID, 변경할 파라미터 및 파라미터의 새 값을 지정합니다. 다음 예제에서는 my-cluster라는 클러스터의 유지 관리 기간을 변경하고 변경 사항을 즉시 적용합니다.

Important

새로운 엔진 버전으로 업그레이드할 수 있습니다. 이에 대한 자세한 내용은 [엔진 버전 업그레이드 \(p. 29\)](#) 단원을 참조하십시오. 하지만 기존의 클러스터를 삭제하고 새로 만들지 않는 한 이전 엔진 버전으로 다운그레이드할 수 없습니다.

읽기 쉽도록 줄바꿈이 추가됩니다.

```
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=AWS4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150901T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

ApplyImmediately 파라미터는 노드 유형, 엔진 버전의 수정 및 클러스터의 노드 수 변경에만 적용됩니다. 이 변경 사항을 즉시 적용하려면 ApplyImmediately 파라미터를 true로 설정하십시오. 다음 번 유지 관리 기간으로 이 변경을 연기하려면 ApplyImmediately 파라미터를 false로 설정하십시오. 유지 관리 기간 변경과 같은 다른 수정은 즉시 적용됩니다.

자세한 내용은 ElastiCache API 참조 항목 [ModifyCacheCluster](#)을 참조하십시오.

클러스터 재부팅

일부 변경 사항은 클러스터를 재부팅해야 적용됩니다. 예를 들어, 일부 파라미터는 파라미터 그룹의 파라미터 값을 변경할 경우 재부팅해야 변경 사항이 적용됩니다.

클러스터를 재부팅할 때 클러스터가 모든 데이터를 플러시하고 해당 엔진을 다시 시작합니다. 이 프로세스 중에는 클러스터에 액세스할 수 없습니다. 클러스터가 해당 데이터를 모두 플러시하기 때문에 클러스터가 다시 사용 가능한 상태가 되면 빈 클러스터로 시작됩니다.

ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 클러스터를 재부팅할 수 있습니다. ElastiCache 콘솔, AWS CLI 또는 ElastiCache API 중 어느 것을 사용하건 단일 클러스터 재부팅만 시작할 수 있습니다. 여러 클러스터를 재부팅하려면 프로세스나 작업에서 반복해야 합니다.

AWS Management 콘솔 사용

ElastiCache 콘솔을 사용하여 클러스터를 재부팅할 수 있습니다.

클러스터를 재부팅하려면 (console)

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 상단 오른쪽 모서리의 목록에서 원하는 AWS 리전을 선택합니다.
3. 재부팅하려는 클러스터에서 실행 중인 엔진을 탐색 창에서 선택합니다.
선택한 엔진을 실행하는 모든 클러스터의 목록이 표시됩니다.
4. 클러스터 이름 왼쪽의 상자를 선택하여 재부팅할 클러스터를 선택합니다.
[Reboot] 버튼이 활성화됩니다.
클러스터를 2개 이상 선택하면 [Reboot] 버튼이 비활성화됩니다.
5. [Reboot]를 선택합니다.
클러스터 재부팅 확인 화면이 나타납니다.
6. 클러스터를 재부팅하려면 [Reboot]를 선택합니다. 클러스터 상태가 [rebooting cluster nodes]로 변경됩니다.
클러스터를 재부팅하지 않으려면 [Cancel]을 선택합니다.

여러 클러스터를 재부팅하려면 재부팅할 클러스터마다 2단계에서 5단계까지 반복합니다. 한 클러스터의 재부팅이 완료되기를 기다렸다가 그다음 클러스터를 재부팅할 필요는 없습니다.

AWS CLI 사용

클러스터 (AWS CLI)를 재부팅하려면 `reboot-cache-cluster` CLI 작업을 사용하십시오.

클러스터의 특정 노드를 재부팅하려면 `--cache-node-ids-to-reboot`를 사용하여 재부팅할 특정 클러스터를 나열하십시오. 다음 명령을 통해 `my-cluster`의 노드 0001, 0002 및 0004가 재부팅됩니다.

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

클러스터의 모든 노드를 재부팅하려면 `--cache-node-ids-to-reboot` 파라미터를 사용하고 클러스터의 모든 노드 ID를 나열하십시오. 자세한 내용은 [reboot-cache-cluster](#)를 참조하십시오.

ElastiCache API 사용

ElastiCache API를 사용하여 클러스터를 재부팅하려면 `RebootCacheCluster` 작업을 사용하십시오.

클러스터의 특정 노드를 재부팅하려면 CacheNodeIdsToReboot를 사용하여 재부팅할 특정 클러스터를 나열하십시오. 다음 명령을 통해 my-cluster의 노드 0001, 0002 및 0004가 재부팅됩니다.

```
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>
```

클러스터의 모든 노드를 재부팅하려면 CacheNodeIdsToReboot 파라미터를 사용하고 클러스터의 모든 노드 ID를 나열하십시오. 자세한 내용은 [RebootCacheCluster](#)를 참조하십시오.

클러스터에 노드 추가

Memcached 클러스터에 노드를 추가하면 클러스터의 파티션 수가 증가합니다. 클러스터의 파티션 수를 변경할 때는 키스페이스의 일부를 다시 매핑해야 정확한 노드에 매핑됩니다. 키스페이스를 일시적으로 다시 매핑하면 클러스터의 캐시 누락 수가 증가합니다. 자세한 내용은 [효율적인 로드 밸런싱을 위해 ElastiCache 클라이언트 구성 \(p. 25\)](#) 단원을 참조하십시오.

ElastiCache 관리 콘솔 AWS CLI 또는 ElastiCache API를 사용하여 클러스터에 노드를 추가할 수 있습니다.

AWS Management 콘솔 사용

항목

- [클러스터 \(console\)에 노드를 추가하려면 \(p. 79\)](#)

클러스터 (console)에 노드를 추가하려면

다음 절차에 따라 클러스터에 노드를 추가할 수 있습니다.

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 노드를 추가하려는 클러스터에서 실행 중인 엔진을 탐색 창에서 선택합니다.

선택한 엔진을 실행하는 클러스터 목록이 표시됩니다.
3. 클러스터 목록에서 노드를 추가하고자 하는 클러스터의 이름 왼쪽에 있는 상자가 아닌 클러스터의 이름을 선택합니다.
4. [Add node]를 선택합니다.
5. Add Node(노드 추가) 대화 상자에 요청을 받은 정보를 입력합니다.
6. 이 노드를 즉시 추가하려면 [Apply Immediately - Yes] 버튼을 선택하고, 클러스터의 다음 유지 관리 기간 중에 이 노드를 추가하려면 [No]를 선택합니다.

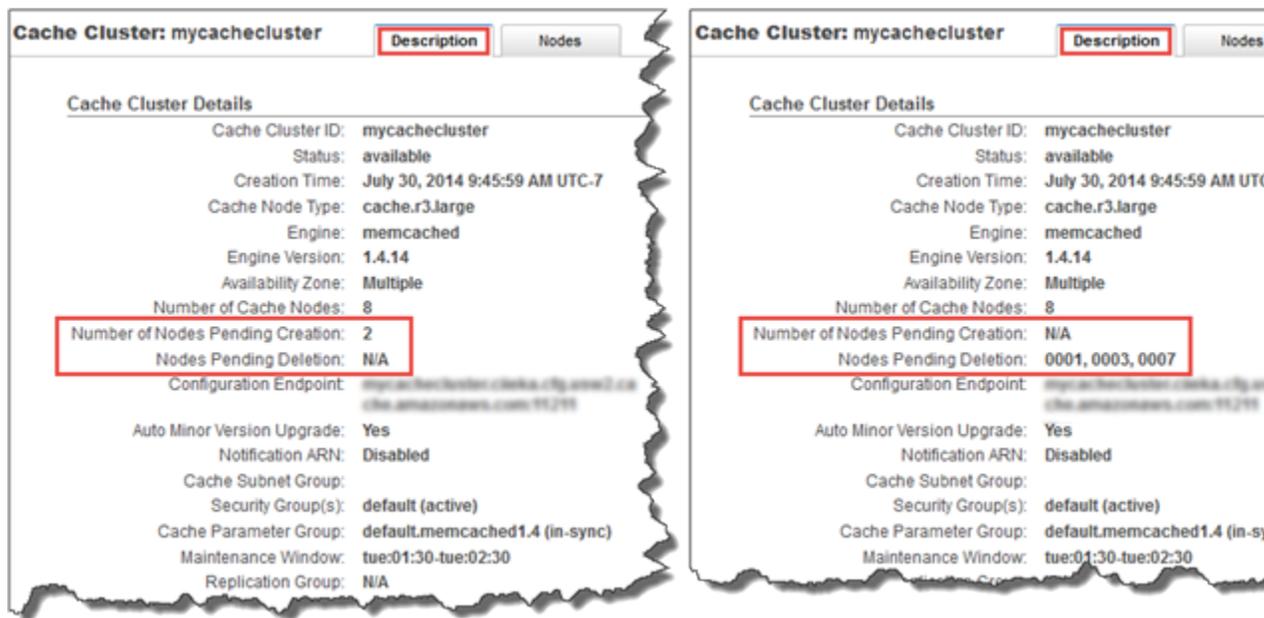
신규 추가 및 제거 요청이 대기 중 요청에 미치는 영향

시나리오	대기 중 작업	신규 요청	Results
시나리오 1	삭제	삭제	신규 삭제 요청(대기 중 또는 즉시)은 대기 중 삭제 요청을 대체합니다.

Amazon ElastiCache ElastiCache
for Memcached 사용 설명서
클러스터에 노드 추가

시나리오	대기 중 작업	신규 요청	Results
			예를 들어 노드 0001, 0003 및 0007에서 삭제 요청이 대기 중일 때 노드 0002 및 0004를 삭제하는 신규 요청이 생성될 경우 노드 0002 및 0004만 삭제됩니다. 노드 0001, 0003 및 0007은 삭제되지 않습니다.
시나리오 2	삭제	생성	신규 생성 요청(대기 중 또는 즉시)은 대기 중 삭제 요청을 대체합니다. 예를 들어 노드 0001, 0003 및 0007에서 삭제 요청이 대기 중일 때 노드를 생성하는 신규 요청이 생성될 경우 새 노드가 생성되고 노드 0001, 0003 및 0007은 삭제되지 않습니다.
시나리오 3	생성	삭제	신규 삭제 요청(대기 중 또는 즉시)은 대기 중 생성 요청을 대체합니다. 예를 들어 노드 2개를 생성하는 요청이 대기 중일 때 노드 0003을 삭제하는 요청이 생성될 경우 새 노드는 생성되지 않고 노드 0003이 삭제됩니다.
시나리오 4	생성	생성	신규 생성 요청은 대기 중 생성 요청에 추가됩니다. 예를 들어 노드 2개를 생성하는 요청이 대기 중일 때 노드 3개를 생성하는 신규 요청이 생성될 경우 신규 요청이 대기 중 요청에 추가되어 노드 5개가 생성됩니다. Important 신규 생성 요청이 Apply Immediately - Yes(즉시 적용 - 예)로 설정된 경우 모든 생성 요청이 즉시 실행됩니다. 신규 생성 요청이 Apply Immediately - No(즉시 적용 - 아니요)로 설정된 경우 모든 생성 요청은 대기 중 작업입니다.

어떤 작업이 대기 중인지 알아보려면 설명 탭을 선택하여 대기 중 생성 또는 삭제 작업이 몇 개인지 확인합니다. 대기 중 생성 작업과 대기 중 삭제 작업이 동시에 있을 수는 없습니다.



7. [Add] 버튼을 선택합니다.

잠시 후 [creating] 상태로 새로운 노드가 노드 목록에 나타납니다. 노드가 표시되지 않으면 브라우저 페이지를 새로 고치십시오. 노드가 사용 가능 상태가 되면 새로운 노드를 사용할 수 있습니다.

AWS CLI 사용

AWS CLI를 사용하여 클러스터에 노드를 추가하려면 다음 파라미터와 함께 AWS CLI 작업 `modify-cache-cluster`를 사용하십시오.

- `--cache-cluster-id` 노드를 추가할 캐시 클러스터의 ID입니다.
- `--num-cache-nodes` `--num-cache-nodes` 파라미터는 수정이 적용된 후 이 클러스터에 포함할 노드 수를 지정합니다. 이 클러스터에 노드를 추가하려면 `--num-cache-nodes`가 이 클러스터의 현재 노드 수보다 커야 합니다. 이 값이 현재 노드 수보다 작으면 ElastiCache가 파라미터 `cache-node-ids-to-remove` 및 클러스터에서 제거할 노드 목록을 필요로 합니다. 자세한 내용은 [AWS CLI 사용 \(p. 85\)](#) 단원을 참조하십시오.
- `--apply-immediately` 또는 `--no-apply-immediately` 이 노드를 즉시 추가할지 아니면 다음 번 유지 관리 기간에 추가할지 지정합니다.

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

이 작업은 다음과 비슷한 출력을 생성합니다(JSON 형식).

```
{
  "CacheCluster": {
    "Engine": "memcached",
    "CacheParameterGroup": {
      "CacheNodeIdsToReboot": [],
      "CacheParameterGroupName": "default.memcached1.4",
      "ParameterApplyStatus": "in-sync"
    },
    "CacheClusterId": "my-cluster",
    "PreferredAvailabilityZone": "us-west-2b",
    "ConfigurationEndpoint": {
      "Port": 11211,
      "Address": "r1h-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"
  }
}
```

자세한 내용은 AWS CLI 항목 [modify-cache-cluster](#)를 참조하십시오.

ElastiCache API 사용

클러스터 (ElastiCache API)에 노드를 추가하려면

- 다음 파라미터와 함께 `ModifyCacheCluster` API 작업을 호출합니다.
 - `CacheClusterId` 노드를 추가할 클러스터의 ID입니다.
 - `NumCacheNodes` `NumCachNodes` 파라미터는 수정이 적용된 후 이 클러스터에 포함할 노드 수를 지정합니다. 이 클러스터에 노드를 추가하려면 `NumCacheNodes`가 이 클러스터의 현재 노드 수보다 커야 합니다. 이 값이 현재 노드 수보다 작으면 ElastiCache가 클러스터에서 제거할 노드 목록과 함께 파라미터 `CacheNodeIdsToRemove`를 필요로 합니다([ElastiCache API 사용 \(p. 86\)](#) 참조).
 - `ApplyImmediately` 이 노드를 즉시 추가할지 아니면 다음 번 유지 관리 기간에 추가할지 지정합니다.
 - `Region` 노드를 추가할 클러스터의 AWS 리전을 지정합니다.

다음 예제에서는 클러스터에 노드를 추가하기 위한 호출을 보여줍니다.

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=AWS4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

자세한 내용은 ElastiCache API 항목 [ModifyCacheCluster](#)를 참조하십시오.

클러스터에서 노드 제거

Memcached 클러스터의 노드 수를 변경할 때마다 키스페이스의 일부라도 다시 매핑해야 정확한 노드에 매핑됩니다. Memcached 클러스터의 로드 밸런싱에 대한 자세한 내용은 [효율적인 로드 밸런싱을 위해 ElastiCache 클라이언트 구성 \(p. 25\)](#) 단원을 참조하십시오.

AWS Management 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 클러스터에서 노드를 삭제할 수 있습니다.

주제

- [AWS Management 콘솔 사용 \(p. 83\)](#)
- [AWS CLI 사용 \(p. 85\)](#)
- [ElastiCache API 사용 \(p. 86\)](#)

AWS Management 콘솔 사용

클러스터 (console)에서 노드를 제거하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 상단 오른쪽 모서리의 목록에서 노드를 제거할 클러스터의 AWS 리전을 선택합니다.
3. 노드를 제거하려는 클러스터에서 실행 중인 엔진을 탐색 창에서 선택합니다.

선택한 엔진을 실행하는 클러스터 목록이 표시됩니다.
4. 클러스터 목록에서 노드를 제거할 클러스터 이름을 선택합니다.

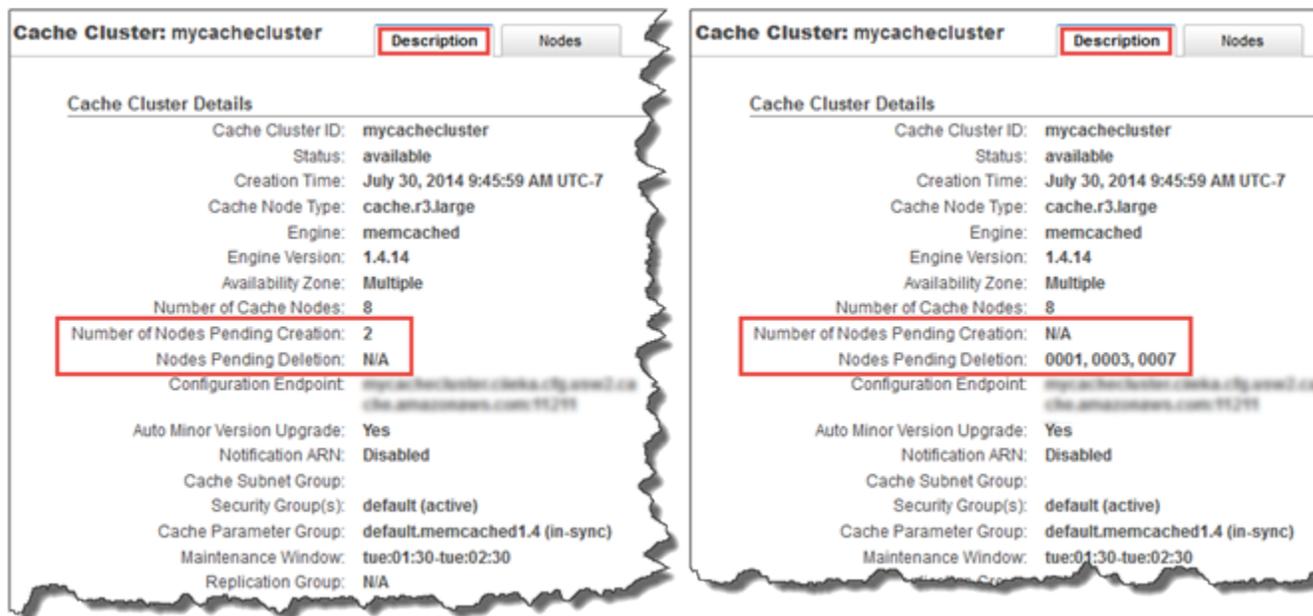
클러스터의 노드 목록이 나타납니다.
5. 제거할 노드의 노드 ID 왼쪽에 있는 상자를 선택합니다. ElastiCache 콘솔을 사용하면 노드를 한 번에 하나만 삭제할 수 있으므로 노드를 여러 개 선택하면 노드 삭제 버튼이 비활성화됩니다.

[Delete Node] 대화 상자가 나타납니다.
6. 노드를 삭제하려면 [Delete Node] 대화 상자를 완료하고 [Delete Node]를 선택합니다. 노드를 삭제하지 않으려면 [Cancel]을 선택합니다.

신규 추가 및 제거 요청이 대기 중 요청에 미치는 영향

시나리오	대기 중 작업	신규 요청	Results
시나리오 1	삭제	삭제	<p>신규 삭제 요청(대기 중 또는 즉시)은 대기 중 삭제 요청을 대체합니다.</p> <p>예를 들어 노드 0001, 0003 및 0007에서 삭제 요청이 대기 중일 때 노드 0002 및 0004를 삭제하는 신규 요청이 생성될 경우 노드 0002 및 0004만 삭제됩니다. 노드 0001, 0003 및 0007은 삭제되지 않습니다.</p>
시나리오 2	삭제	생성	<p>신규 생성 요청(대기 중 또는 즉시)은 대기 중 삭제 요청을 대체합니다.</p> <p>예를 들어 노드 0001, 0003 및 0007에서 삭제 요청이 대기 중일 때 노드를 생성하는 신규 요청이 생성될 경우 새 노드가 생성되고 노드 0001, 0003 및 0007은 삭제되지 않습니다.</p>
시나리오 3	생성	삭제	<p>신규 삭제 요청(대기 중 또는 즉시)은 대기 중 생성 요청을 대체합니다.</p> <p>예를 들어 노드 2개를 생성하는 요청이 대기 중일 때 노드 0003을 삭제하는 요청이 생성될 경우 새 노드는 생성되지 않고 노드 0003이 삭제됩니다.</p>
시나리오 4	생성	생성	<p>신규 생성 요청은 대기 중 생성 요청에 추가됩니다.</p> <p>예를 들어 노드 2개를 생성하는 요청이 대기 중일 때 노드 3개를 생성하는 신규 요청이 생성될 경우 신규 요청이 대기 중 요청에 추가되어 노드 5개가 생성됩니다.</p> <p>Important</p> <p>신규 생성 요청이 Apply Immediately - Yes(즉시 적용 - 예)로 설정된 경우 모든 생성 요청이 즉시 실행됩니다. 신규 생성 요청이 Apply Immediately - No(즉시 적용 - 아니요)로 설정된 경우 모든 생성 요청은 대기 중 작업입니다.</p>

어떤 작업이 대기 중인지 알아보려면 설명 탭을 선택하여 대기 중 생성 또는 삭제 작업이 몇 개인지 확인합니다. 대기 중 생성 작업과 대기 중 삭제 작업이 동시에 있을 수는 없습니다.



AWS CLI 사용

1. 제거할 노드의 ID를 확인합니다. 자세한 내용은 [클러스터 세부 정보 보기 \(p. 72\)](#) 단원을 참조하십시오.
2. 다음 예제와 같이 제거할 노드 목록과 함께 `modify-cache-cluster` CLI 작업을 사용하십시오.

명령줄 인터페이스를 사용하여 클러스터에서 노드를 제거하려면 다음 파라미터와 함께 `modify-cache-cluster` 명령을 사용하십시오.

- `--cache-cluster-id` 노드를 제거할 캐시 클러스터의 ID입니다.
- `--num-cache-nodes` `--num-cache-nodes` 파라미터는 수정이 적용된 후 이 클러스터에 포함할 노드 수를 지정합니다.
- `--cache-node-ids-to-remove` 이 클러스터에서 제거할 노드 ID 목록입니다.
- `--apply-immediately` 또는 `--no-apply-immediately` 이 노드를 즉시 제거할지 아니면 다음 번 유지 관리 기간에 제거할지 지정합니다.
- `--region` 노드를 제거할 클러스터의 리전을 지정합니다.

다음 예제에서는 `my-cluster` 클러스터에서 노드 0001을 즉시 제거합니다.

Linux, macOS, or Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

이 작업은 다음과 비슷한 출력을 생성합니다(JSON 형식).

```
{  
  "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"  
  }  
}
```

자세한 내용은 AWS CLI 항목 [describe-cache-cluster](#) 및 [modify-cache-cluster](#)를 참조하십시오.

ElastiCache API 사용

ElastiCache API를 사용하여 노드를 제거하려면 다음과 같이 캐시 클러스터 ID 및 제거할 노드 목록과 함께 `ModifyCacheCluster` API 작업을 호출하십시오.

- `CacheClusterId` 노드를 제거할 캐시 클러스터의 ID입니다.
- `NumCacheNodes` `NumCacheNodes` 파라미터는 수정이 적용된 후 이 클러스터에 포함할 노드 수를 지정합니다.
- `CacheNodeIdsToRemove.member.n` 클러스터에서 제거할 노드 ID 목록입니다.
 - `CacheNodeIdsToRemove.member.1=0004`
 - `CacheNodeIdsToRemove.member.1=0005`

- `ApplyImmediately` 이 노드를 즉시 제거할지 아니면 다음 번 유지 관리 기간에 제거할지 지정합니다.
- `Region` 노드를 제거할 클러스터의 리전을 지정합니다.

다음 예제에서는 `my-cluster` 클러스터에서 노드 0004 및 0005를 즉시 제거합니다.

```
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=AWS4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

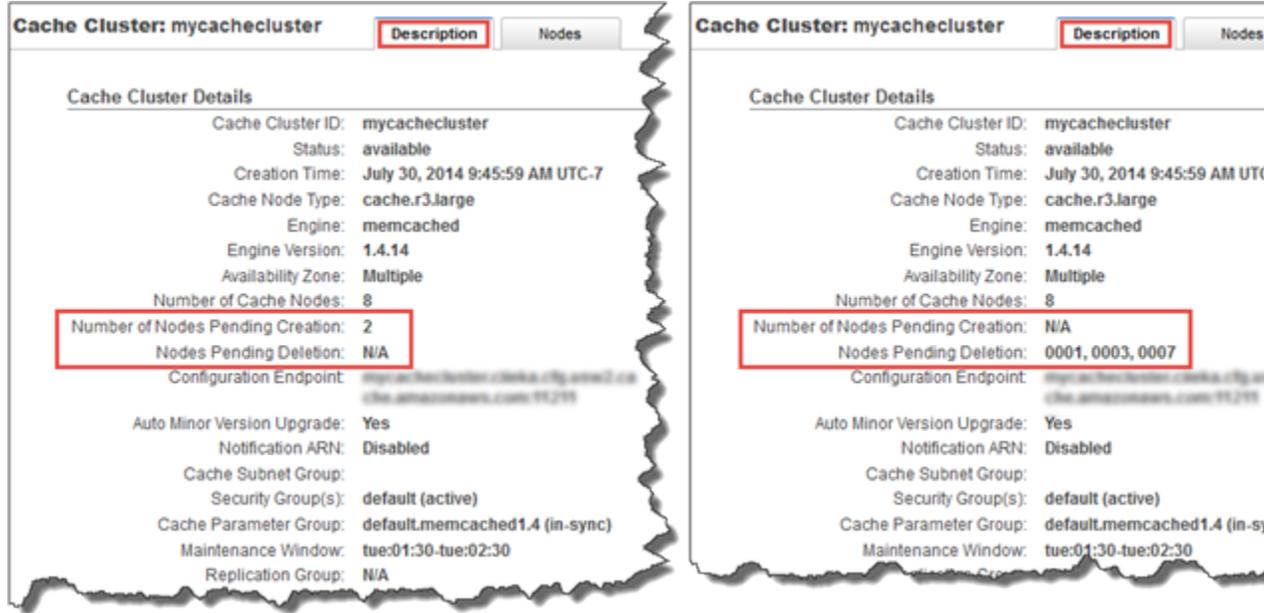
자세한 내용은 ElastiCache API 항목 [ModifyCacheCluster](#)를 참조하십시오.

대기 중인 노드 추가 또는 삭제 작업 취소

변경 사항을 즉시 적용하지 않도록 선택하면 다음 번 유지 관리 기간에 수행할 때까지 작업이 [pending] 상태가 됩니다. 대기 중인 작업을 취소할 수 있습니다.

대기 중인 작업을 취소하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 상단 오른쪽 모서리의 목록에서 대기 중인 노드 추가 또는 삭제 작업을 취소할 AWS 리전을 선택합니다.
3. 취소하려는 대기 중 작업이 있는 클러스터에서 실행 중인 엔진을 탐색 창에서 선택합니다. 선택한 엔진을 실행하는 모든 클러스터의 목록이 표시됩니다.
4. 클러스터 목록에서 보류 중인 작업을 취소하고자 하는 클러스터의 이름 왼쪽에 있는 상자가 아닌 클러스터의 이름을 선택합니다.
5. 어떤 작업이 대기 중인지 알아보려면 설명 탭을 선택하여 대기 중 생성 또는 삭제 작업이 몇 개인지 확인합니다. 대기 중 생성 작업과 대기 중 삭제 작업이 동시에 있을 수는 없습니다.



6. [Nodes] 탭을 선택합니다.
7. 대기 중인 작업을 모두 취소하려면 [Cancel Pending]을 클릭합니다. [Cancel Pending] 대화 상자가 나타납니다.
8. [Cancel Pending] 버튼을 선택하여 대기 중인 작업을 모두 취소하도록 확인하거나 [Cancel]을 선택하여 작업을 유지합니다.

클러스터 삭제하기

클러스터가 available 상태면 클러스터를 적극 사용하고 있는지 여부에 관계없이 요금이 부과됩니다. 요금 발생을 중지하려면 클러스터를 삭제하십시오.

Warning

Redis용 ElastiCache 클러스터를 삭제하는 경우 수동 스냅샷은 보존됩니다. 클러스터를 삭제하기 전에 최종 스냅샷을 생성할 수 있는 옵션도 있습니다. 자동 캐시 스냅샷은 보존되지 않습니다.

AWS Management 콘솔 사용

다음은 배포에서 클러스터 하나를 삭제하는 절차입니다. 클러스터를 여러 개 삭제하려면 삭제할 클러스터마다 절차를 반복하십시오. 클러스터 하나를 다 삭제한 후 다른 클러스터 삭제 절차가 시작될 때까지 기다릴 필요는 없습니다.

클러스터를 삭제하려면

1. AWS Management 콘솔에 로그인한 후 <https://console.aws.amazon.com/elasticache/>에서 Amazon ElastiCache 콘솔을 엽니다.
2. ElastiCache 콘솔 대시보드에서 삭제하려는 클러스터가 실행 중인 엔진을 선택합니다.
선택한 엔진을 실행 중인 모든 클러스터의 목록이 표시됩니다.
3. 삭제할 클러스터를 선택하려면 클러스터 목록에서 클러스터의 이름을 선택합니다.

Important

ElastiCache 콘솔에서는 클러스터를 한 번에 하나씩만 삭제할 수 있습니다. 여러 클러스터를 선택하면 삭제 작업이 비활성화됩니다.

4. [Actions] 버튼을 선택하고 작업 목록에서 [Delete]를 선택합니다.
5. 클러스터 삭제 확인 화면에서 삭제를 선택하여 클러스터를 삭제하거나 취소를 선택하여 클러스터를 유지합니다.

[Delete]를 선택한 경우 클러스터 상태가 [deleting]으로 바뀝니다.

클러스터가 클러스터 목록에서 제거되는 즉시 요금 부과가 중단됩니다.

AWS CLI 사용

다음 코드는 캐시 클러스터 `my-cluster`를 삭제합니다.

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

`delete-cache-cluster` CLI 작업은 캐시 클러스터를 하나만 삭제합니다. 캐시 클러스터를 여러 개 삭제하려면 삭제할 캐시 클러스터마다 `delete-cache-cluster`를 호출하십시오. 캐시 클러스터 하나를 다 삭제한 후 다른 캐시 클러스터를 삭제할 때까지 기다릴 필요는 없습니다.

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

자세한 내용은 AWS CLI의 ElastiCache 항목 [delete-cache-cluster](#)를 참조하십시오.

ElastiCache API 사용

다음 코드는 클러스터 `my-cluster`를 삭제합니다.

```
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=AWS4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150202T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

`DeleteCacheCluster` API 작업은 캐시 클러스터를 하나만 삭제합니다. 캐시 클러스터를 여러 개 삭제하려면 삭제할 캐시 클러스터마다 `DeleteCacheCluster`를 호출하십시오. 캐시 클러스터 하나를 다 삭제한 후 다른 캐시 클러스터를 삭제할 때까지 기다릴 필요는 없습니다.

자세한 내용은 ElastiCache API 참조 항목 [DeleteCacheCluster](#)을 참조하십시오.

클러스터에 액세스

Amazon ElastiCache 인스턴스는 Amazon EC2 인스턴스를 통해서만 액세스하도록 설계되었습니다.

Amazon Virtual Private Cloud(Amazon VPC)에서 ElastiCache 인스턴스를 시작하면 동일한 Amazon VPC의 Amazon EC2 인스턴스에서 ElastiCache 인스턴스에 액세스할 수 있습니다. 또는 VPC 피어링을 사용하여 다른 Amazon VPC의 Amazon EC2에서 ElastiCache 인스턴스에 액세스할 수 있습니다.

EC2 Classic에서 ElastiCache 인스턴스를 시작하면, 인스턴스와 연결된 Amazon EC2 보안 그룹의 캐시 보안 그룹에 대한 액세스를 허용하여 EC2 인스턴스가 클러스터에 액세스하는 것을 허용합니다. 기본적으로 클러스터에 대한 액세스는 클러스터를 시작한 계정으로 제한됩니다.

주제

- 클러스터의 플랫폼 확인 (p. 90)
- 클러스터에 액세스 권한 부여 (p. 92)

클러스터의 플랫폼 확인

계속하기 전에 EC2-VPC로 시작했는지, EC2-Classic으로 시작했는지 확인합니다.

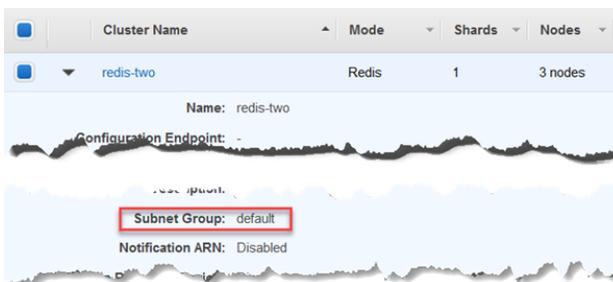
자세한 정보는 [Detecting Your Supported Platforms and Whether You Have a Default VPC](#)를 참조하십시오.

ElastiCache 콘솔을 사용하여 클러스터 플랫폼 확인

다음 절차에서는 ElastiCache 콘솔을 사용하여 클러스터를 EC2-VPC로 시작했는지, EC2-Classic으로 시작했는지 확인합니다.

ElastiCache 콘솔을 사용하여 클러스터 플랫폼을 확인하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. Memcached 엔진을 실행 중인 클러스터의 목록을 보려면 왼쪽 탐색 창에서 Memcached를 선택합니다.
3. 클러스터 목록에서 클러스터 이름 왼쪽의 상자를 선택하여 액세스 권한을 부여할 클러스터를 확장합니다.
4. [Subnet group:]을 찾습니다.



- [Subnet group]에 이름이 있는 경우에는 여기 나와 있듯이 EC2-VPC에서 클러스터를 시작한 것이므로 클러스터를 EC2-VPC로 시작한 경우 (p. 92)의 지침을 따라야 합니다.
- [Subnet group] 이름 대신 대시(-)가 있는 경우에는 EC2-Classic에서 클러스터를 시작한 것이므로 EC2-Classic에서 실행 중인 클러스터를 시작한 경우 (p. 92)의 지침을 따라야 합니다.

자세한 정보는 [Detecting Your Supported Platforms and Whether You Have a Default VPC](#)를 참조하십시오.

AWS CLI를 사용하여 클러스터 플랫폼 확인

다음 절차에서는 AWS CLI를 사용하여 클러스터를 EC2-VPC로 시작했는지, EC2-Classic으로 시작했는지 확인합니다.

AWS CLI를 사용하여 클러스터 플랫폼을 확인하려면

1. 명령 창을 엽니다.
2. 명령 프롬프트에서 다음 명령을 실행합니다.

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

이 명령의 JSON 출력은 다음과 같습니다. 공간 절약을 위해 일부 출력이 생략됩니다.

```
{
  "CacheClusters": [
    {
      "Engine": "memcached",
      "AuthTokenEnabled": false,
      "CacheParameterGroup": {
        "CacheNodeIdsToReboot": [],
        "CacheParameterGroupName": "default.memcached1.4",
        "ParameterApplyStatus": "in-sync"
      },
      "CacheClusterId": "my-cluster-001",
      "CacheSecurityGroups": [],
      "NumCacheNodes": 1,
      "AtRestEncryptionEnabled": false,
      "CacheClusterCreateTime": "2018-01-16T20:09:34.449Z",
      "ReplicationGroupId": "my-cluster",
      "AutoMinorVersionUpgrade": true,
      "CacheClusterStatus": "available",
      "PreferredAvailabilityZone": "us-east-2a",
      "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
      "SecurityGroups": [
        {
          "Status": "active",
          "SecurityGroupId": "sg-e8c03081"
        }
      ],
      "TransitEncryptionEnabled": false,
      "CacheSubnetGroupName": "default",
      "EngineVersion": "3.2.10",
      "PendingModifiedValues": {},
      "PreferredMaintenanceWindow": "sat:05:30-sat:06:30",
      "CacheNodeType": "cache.t2.medium"
    }
  ]
}
```

```
}
```

- CacheSubnetGroupName 값이 있는 경우에는 EC2-VPC에서 클러스터를 시작한 것이므로 **클러스터를 EC2-VPC로 시작한 경우 (p. 92)**의 지침을 따라야 합니다.
- CacheSubnetGroupName 값이 없는 경우에는 EC2-Classic에서 클러스터를 시작한 것이므로 **EC2-Classic에서 실행 중인 클러스터를 시작한 경우 (p. 92)**의 지침을 따라야 합니다.

클러스터에 액세스 권한 부여

클러스터를 EC2-VPC로 시작한 경우

Amazon Virtual Private Cloud(Amazon VPC)로 클러스터를 시작한 경우 동일한 Amazon VPC에서 실행 중인 Amazon EC2 인스턴스에서만 ElastiCache 클러스터에 연결할 수 있습니다. 이 경우 클러스터에 네트워크 진입을 허용해야 합니다.

Amazon VPC 보안 그룹에서 클러스터로의 네트워크 진입을 허용하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/ec2/>에서 Amazon EC2 콘솔을 엽니다.
2. 탐색 창의 [Network & Security] 아래에서 [Security Groups]를 선택합니다.
3. 보안 그룹 목록에서 Amazon VPC를 위한 보안 그룹을 선택합니다. ElastiCache 사용을 위한 보안 그룹을 생성하지 않는 한 이 보안 그룹의 이름은 default로 지정됩니다.
4. [Inbound] 탭을 선택하고 다음을 수행합니다.
 - a. [Edit]를 선택합니다.
 - b. [Add another rule]을 선택합니다.
 - c. [Type] 열에서 [Custom TCP rule]을 선택합니다.
 - d. [Port range] 상자에 클러스터 노드의 포트 번호를 입력합니다. 이 번호는 클러스터를 시작할 때 지정한 번호와 동일해야 합니다. Memcached의 기본 포트는 **11211**입니다.
 - e. 소스 상자에서 포트 범위(0.0.0.0/0)를 가진 위치 무관을 선택하면 Amazon VPC 내에서 시작한 Amazon EC2 인스턴스를 ElastiCache 노드에 연결할 수 있습니다.

Important

ElastiCache 클러스터를 0.0.0.0/0(4.e.단계)으로 열면 공용 IP 주소가 없기 때문에 클러스터가 인터넷에 노출되지 않으므로 VPC 외부에서 액세스할 수 없습니다. 그러나 기본 보안 그룹이 고객 계정의 다른 Amazon EC2 인스턴스에 적용될 수 있으며 이러한 인스턴스는 공용 IP 주소를 가질 수 있습니다. 포트 6379에서 무언가를 실행하면 비의도적으로 해당 서비스가 노출될 수 있습니다. 따라서 ElastiCache가 독점적으로 사용하는 VPC 보안 그룹을 생성하는 것이 좋습니다. 자세한 정보는 [사용자 지정 보안 그룹](#)을 참조하십시오.

- f. Save를 선택합니다.

Amazon EC2 인스턴스를 Amazon VPC로 시작하면 해당 인스턴스를 ElastiCache 클러스터에 연결할 수 있습니다.

EC2-Classic에서 실행 중인 클러스터를 시작한 경우

클러스터를 EC2-Classic으로 시작한 경우 Amazon EC2 인스턴스가 클러스터에 액세스할 수 있게 하려면 인스턴스와 연결된 Amazon EC2 보안 그룹을 허용하여 캐시 보안 그룹에 액세스합니다.

클러스터에 Amazon EC2 보안 그룹의 액세스를 허용하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.

2. 보안 그룹 목록을 표시하려면 왼쪽 탐색 창의 [Security Groups]를 선택합니다.

Important

[Security Groups]이 탐색 창에 표시되지 않은 경우에는 EC2-Classic이 아니라 EC2-VPC에서 클러스터를 시작한 것이므로 [클러스터를 EC2-VPC로 시작한 경우 \(p. 92\)](#)의 지침을 따라야 합니다.

3. [default] 보안 그룹의 왼쪽에 있는 확인란을 선택합니다.
4. 화면 아래쪽에 있는 목록에서 승인하려는 [EC2 Security Group Name]을 선택합니다.
5. 액세스를 허가하려면 [Add]를 선택합니다.

보안 그룹과 연결된 Amazon EC2 인스턴스를 ElastiCache 클러스터에 연결하도록 승인했습니다.

보안 그룹의 액세스를 취소하려면 승인한 보안 그룹 목록에서 보안 그룹을 찾은 다음 [Remove]를 선택합니다.

ElastiCache 보안 그룹에 대한 자세한 내용은 [보안 그룹: EC2-Classic \(p. 188\)](#) 단원을 참조하십시오.

AWS 외부에서 ElastiCache 리소스 액세스

ElastiCache는 VPC에 내부적으로 사용하도록 설계된 서비스입니다. 인터넷 트래픽의 지연 시간 및 보안 문제로 인해 외부 액세스는 권장되지 않습니다. 그러나 테스트 또는 개발 목적으로 ElastiCache에 대한 외부 액세스가 필요한 경우 VPN을 통해 수행할 수 있습니다.

AWS 클라이언트 VPN을 사용하면 다음과 같은 이점을 통해 ElastiCache 노드에 대한 외부 액세스를 허용합니다.

- 승인된 사용자 또는 인증 키에 대한 제한된 액세스
- VPN 클라이언트와 AWS VPN 엔드포인트 간의 암호화된 트래픽
- 특정 서브넷 또는 노드에 대한 제한된 액세스
- 사용자 또는 인증 키로부터의 액세스를 쉽게 취소
- 감사 연결

다음 절차에서는 다음 작업을 수행하는 방법을 보여줍니다.

주제

- [인증 기관 생성 \(p. 93\)](#)
- [AWS Client VPN 구성 요소 구성 \(p. 94\)](#)
- [VPN 클라이언트 구성 \(p. 96\)](#)

인증 기관 생성

다양한 기술이나 도구를 사용하여 CA(인증 기관)를 생성할 수 있습니다. [OpenVPN](#) 프로젝트에서 제공하는 `easy-rsa` 유틸리티를 사용하는 것이 좋습니다. 선택한 옵션에 관계없이 키를 안전하게 유지해야 합니다. 다음 절차에서는 `easy-rsa` 스크립트를 다운로드하고 인증 기관과 첫 번째 VPN 클라이언트를 인증하는 키를 생성합니다.

- 초기 인증서를 생성하려면 터미널을 열고 다음 작업을 수행하십시오.
 - `git clone https://github.com/OpenVPN/easy-rs`
 - `cd easy-rsa`
 - `./easyrsa3/easyrsa init-pki`
 - `./easyrsa3/easyrsa build-ca nopass`
 - `./easyrsa3/easyrsa build-server-full server nopass`

```
• ./easyrsa3/easyrsa build-client-full client1.domain.tld nopass
```

인증서를 포함하는 pki 하위 디렉터리는 easy-rsa 아래에 생성됩니다.

- 다음과 같이 AWS Certificate Manager(ACM)에 서버 인증서를 제출합니다.
 - ACM 콘솔에서 인증서 관리자를 선택합니다.
 - 인증서 가져오기를 선택합니다.
 - easy-rsa/pki/issued/server.crt 파일에서 사용할 수 있는 퍼블릭 키 인증서를 인증서 본문 필드에 입력합니다.
 - easy-rsa/pki/private/server.key에서 사용할 수 있는 프라이빗 키를 인증서 프라이빗 키 필드에 붙여 넣습니다. BEGIN AND END PRIVATE KEY 사이의 모든 선(BEGIN 및 END 선 포함)을 선택해야 합니다.
 - easy-rsa/pki/ca.crt 파일에서 사용할 수 있는 CA 퍼블릭 키를 인증서 체인 필드에 붙여넣습니다.
 - 검토 및 가져오기를 선택합니다.
 - 가져오기를 선택합니다.

AWS CLI를 사용하여 서버의 인증서를 ACM에 제출하려면 다음 명령을 실행하십시오. aws acm import-certificate --certificate file://easy-rsa/pki/issued/server.crt --private-key file://easy-rsa/pki/private/server.key --certificate-chain file://easy-rsa/pki/ca.crt --region *region*

나중에 사용할 수 있도록 인증서 ARN을 기록해 둡니다.

AWS Client VPN 구성 요소 구성

AWS 콘솔 사용

AWS 콘솔에서 서비스, VPC 순으로 선택합니다.

Virtual Private Network(가상 프라이빗 네트워크)에서 Client VPN Endpoints(클라이언트 VPN 엔드포인트)를 선택하고 다음을 수행합니다.

AWS Client VPN 구성 요소 구성

- Create Client VPN Endpoint(클라이언트 VPN 엔드포인트 생성)를 선택합니다.
- 다음과 같은 옵션을 지정할 수 있습니다.
 - Client IPv4 CIDR(클라이언트 IPv4 CIDR): 넷마스크가 최소 /22 범위에 있는 프라이빗 네트워크를 사용합니다. 선택한 서브넷이 VPC 네트워크의 주소와 충돌하지 않는지 확인합니다. 예: 10.0.0.0/22.
 - Server certificate ARN(서버 인증서 ARN)에서 앞서 가져온 인증서의 ARN을 선택합니다.
 - Use mutual authentication(상호 인증 사용)을 선택합니다.
 - Client certificate ARN(클라이언트 인증서 ARN)에서 앞서 가져온 인증서의 ARN을 선택합니다.
 - Create Client VPN Endpoint(클라이언트 VPN 엔드포인트 생성)를 선택합니다.

AWS CLI 사용

다음 명령을 실행합니다.

```
aws ec2 create-client-vpn-endpoint --client-cidr-block "10.0.0.0/22" --server-certificate-arn arn:aws:acm:us-east-1:012345678912:certificate/0123abcd-ab12-01a0-123a-123456abcdef --authentication-options Type=certificate-authentication, MutualAuthentication={ClientRootCertificateChainArn=arn:aws:acm:us-east-1:012345678912:certificate/123abcd-ab12-01a0-123a-123456abcdef} --connection-log-options Enabled=false
```

출력 예:

```
"ClientVpnEndpointId": "cvpn-endpoint-0123456789abcdefg",  
"Status": { "Code": "pending-associate" }, "DnsName": "cvpn-  
endpoint-0123456789abcdefg.prod.clientvpn.us-east-1.amazonaws.com" }
```

대상 네트워크를 VPN 엔드포인트에 연결

- 새 VPN 엔드포인트를 선택한 다음 연결 탭을 선택합니다.
- 연결을 선택하고 다음 옵션을 지정합니다.
 - VPC: ElastiCache 클러스터의 VPC를 선택합니다.
 - ElastiCache 클러스터의 네트워크 중 하나를 선택합니다. 의심스러운 경우 ElastiCache 대시보드의 서브넷 그룹에 있는 네트워크를 검토하십시오.
 - 연결을 선택합니다. 필요한 경우 나머지 네트워크에 대해 이 단계를 반복합니다.

AWS CLI 사용

다음 명령을 실행합니다.

```
aws ec2 associate-client-vpn-target-network --client-vpn-endpoint-id cvpn-  
endpoint-0123456789abcdefg --subnet-id subnet-0123456789abcdef
```

출력 예:

```
"Status": { "Code": "associating" }, "AssociationId": "cvpn-  
assoc-0123456789abcdef" }
```

VPN 보안 그룹 검토

VPN 엔드포인트는 VPC의 기본 보안 그룹을 자동으로 채택합니다. 인바운드 및 아웃바운드 규칙을 점검하고 보안 그룹이 VPN 네트워크(VPN 엔드포인트 설정에 정의됨)에서 서비스 포트의 ElastiCache 네트워크로의 트래픽을 허용하는지 확인합니다(기본적으로 Redis의 경우 6379, Memcached의 경우 11211).

VPN 엔드포인트에 할당된 보안 그룹을 변경해야 하는 경우 다음과 같이 진행합니다.

- 현재 보안 그룹을 선택합니다.
- Apply Security Group(보안 그룹 적용)을 선택합니다.
- 보안 그룹을 선택합니다.

AWS CLI 사용

다음 명령을 실행합니다.

```
aws ec2 apply-security-groups-to-client-vpn-target-network --client-vpn-  
endpoint-id cvpn-endpoint-0123456789abcdefga --vpc-id vpc-0123456789abcdef --  
security-group-ids sg-0123456789abcdef
```

출력 예:

```
"SecurityGroupIds": [ "sg-0123456789abcdef" ] }
```

Note

ElastiCache 보안 그룹은 VPN 클라이언트에서 오는 트래픽을 허용해야 합니다. 클라이언트의 주소는 VPC 네트워크에 따라 VPN 엔드포인트 주소로 마스킹 처리됩니다. 따라서 ElastiCache 보안 그룹에서 인바운드 규칙을 생성할 때 VPC 네트워크(VPN 클라이언트 네트워크가 아님)를 고려하십시오.

대상 네트워크에 대한 VPN 액세스 승인

Authorization(권한 부여) 탭에서 Authorize Ingress(권한 부여 승인)를 선택하고 다음과 같이 지정합니다.

- 액세스를 활성화할 대상 네트워크: 0.0.0.0/0을 사용하여 모든 네트워크(인터넷 포함)에 대한 액세스를 허용하거나 ElastiCache 네트워크/호스트를 제한합니다.
- Grant access to:(다음에 대한 액세스 권한 부여:)에서 Allow access to all users(모든 사용자에게 액세스 허용)를 선택합니다.
- Add Authorization Rules(권한 부여 규칙 추가)를 선택합니다.

AWS CLI 사용

다음 명령을 실행합니다.

```
aws ec2 authorize-client-vpn-ingress --client-vpn-endpoint-id cvpn-  
endpoint-0123456789abcdefg --target-network-cidr 0.0.0.0/0 --authorize-all-  
groups
```

출력 예:

```
{ "Status": { "Code": "authorizing" } }
```

VPN 클라이언트에서 인터넷에 액세스하도록 허용

VPN을 통해 인터넷을 검색해야 하는 경우 추가 경로를 만들어야 합니다. 라우팅 테이블 탭을 선택한 다음 Create Route(라우팅 생성)를 선택합니다.

- 라우팅 대상 주소: 0.0.0.0/0
- Target VPC Subnet ID(대상 VPC 서브넷 ID): 인터넷 액세스 권한이 있는 연결된 서브넷 중 하나를 선택합니다.
- Create Route(라우팅 생성)를 선택합니다.

AWS CLI 사용

다음 명령을 실행합니다.

```
aws ec2 create-client-vpn-route --client-vpn-endpoint-id cvpn-  
endpoint-0123456789abcdefg --destination-cidr-block 0.0.0.0/0 --target-vpc-  
subnet-id subnet-0123456789abcdef
```

출력 예:

```
{ "Status": { "Code": "creating" } }
```

VPN 클라이언트 구성

AWS Client VPN 대시보드에서 최근에 생성된 VPN 엔드포인트를 선택하고 Download Client Configuration(클라이언트 구성 다운로드)를 선택합니다. 구성 파일과 easy-rsa/pki/private/client1.domain.tld.key 및 easy-rsa/pki/issued/client1.domain.tld.crt 파일을 복사합니다. 구성 파일을 편집하고 다음 파라미터를 변경하거나 추가합니다.

- cert: client1.domain.tld.crt 파일을 가리키는 파라미터 인증서가 있는 새 줄을 추가합니다. 파일의 전체 경로를 사용합니다. 예: cert /home/user/.cert/client1.domain.tld.crt
- cert: key: client1.domain.tld.key 파일을 가리키는 파라미터 키가 있는 새 줄을 추가합니다. 파일의 전체 경로를 사용합니다. 예: key /home/user/.cert/client1.domain.tld.key

다음 명령을 사용하여 VPN 연결을 설정합니다. sudo openvpn --config downloaded-client-config.ovpn

액세스 취소

특정 클라이언트 키의 액세스를 무효화해야 하는 경우 CA에서 키를 취소해야 합니다. 그런 다음 취소 목록을 AWS Client VPN에 제출합니다.

easy-rsa로 키 취소:

- `cd easy-rsa`
 - `./easyrsa3/easyrsa revoke client1.domain.tld`
 - 계속하려면 “예”를 입력하고 중단하려면 다른 값을 입력합니다.
- ```
Continue with revocation: `yes` ... * `./easyrsa3/easyrsa gen-crl
```
- 업데이트된 CRL이 생성되었습니다. CRL 파일: `/home/user/easy-rsa/pki/crl.pem`

취소 목록을 AWS Client VPN으로 가져오기:

- AWS Management 콘솔에서 서비스, VPC 순으로 선택합니다.
- Client VPN Endpoints(클라이언트 VPN 엔드포인트)를 선택합니다.
- 클라이언트 VPN 엔드포인트를 선택한 다음 작업 -> Import Client Certificate CRL(클라이언트 인증서 CRL 가져오기)을 선택합니다.
- `crl.pem` 파일의 내용을 붙여넣습니다.

## AWS CLI 사용

다음 명령을 실행합니다.

```
aws ec2 import-client-vpn-client-certificate-revocation-list --certificate-revocation-list file://./easy-rsa/pki/crl.pem --client-vpn-endpoint-id cvpn-endpoint-0123456789abcdefg
```

출력 예:

```
Example output: { "Return": true }
```

# 글로벌 데이터 스토어를 사용한 AWS 리전 간 복제

RedisDyD 글로벌 데이터 스토어 기능을 사용하면 AWS 리전 간에 빠르고 안정적이며 안전한 완전 관리형 복제를 수행할 수 있습니다. 이 기능을 사용하면 Redis용 ElastiCache에 대한 리전 간 읽기 전용 복제본 클러스터를 생성하여 AWS 리전 간에 지연 시간이 짧은 읽기 및 재해 복구를 수행할 수 있습니다.

다음 섹션에서는 글로벌 데이터 스토어로 작업하는 방법에 대한 설명을 찾을 수 있습니다.

## 주제

- [개요 \(p. 97\)](#)
- [사전 조건 및 제한 사항 \(p. 98\)](#)
- [글로벌 데이터 스토리지 사용\(콘솔\) \(p. 99\)](#)
- [글로벌 데이터 저장소\(CLI\) 사용 \(p. 107\)](#)

## 개요

각 글로벌 데이터 스토어는 서로 복제하는 하나 이상의 클러스터 모음입니다.

글로벌 데이터 스토어는 다음과 같이 구성됩니다.

- 기본(활성) 클러스터 – 기본 클러스터는 글로벌 데이터 스토어 내의 모든 클러스터에 복제되는 쓰기를 허용합니다. 기본 클러스터는 읽기 요청도 허용합니다.
- 보조(수동) 클러스터 – 보조 클러스터는 읽기 요청만 허용하고 기본 클러스터에서 데이터 업데이트를 복제합니다. 보조 클러스터는 기본 클러스터와 다른 AWS 리전에 있어야 합니다.

ElastiCache에서 글로벌 데이터 스토어를 생성하면 Redis용 ElastiCache에서 기본 클러스터에서 보조 클러스터로 데이터를 자동으로 복제합니다. Redis 데이터를 복제할 AWS 리전을 선택한 다음 해당 AWS 리전에 보조 클러스터를 생성합니다. 그런 다음 ElastiCache는 두 클러스터 간의 자동 비동기식 데이터 복제를 설정하고 관리합니다.

Redis용 글로벌 데이터 스토어를 사용하면 다음과 같은 이점을 얻을 수 있습니다.

- 지리적 성능 – 추가 AWS 리전에 원격 복제본 클러스터를 설정하고 이들 간에 데이터를 동기화하면 해당 AWS 리전에서 데이터 액세스 지연 시간을 줄일 수 있습니다. 글로벌 데이터 스토어는 AWS 리전 간에 지연 시간이 짧은 지리적 읽기를 제공하여 애플리케이션의 응답성을 높일 수 있습니다.
- 재해 복구 – 글로벌 데이터 스토어의 기본 클러스터에서 성능 저하가 발생하는 경우 보조 클러스터를 새 기본 클러스터로 승격할 수 있습니다. 보조 클러스터가 포함된 AWS 리전에 연결하여 이를 수행할 수 있습니다.

다음 다이어그램은 글로벌 데이터 스토어가 작동하는 방식을 보여줍니다.

## 사전 조건 및 제한 사항

글로벌 데이터 스토어를 시작하기 전에 다음 사항에 유의하십시오.

- 글로벌 데이터 스토어는 ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-2, eu-west-1, us-east-1, us-east-2, us-west-1, us-west-2 등의 AWS 리전에서 지원됩니다.
- 글로벌 데이터 스토어를 사용하려면 Redis 엔진 버전 5.0.6 이상과 R5 또는 M5 노드 유형을 사용합니다.
- 글로벌 데이터 스토어의 모든 —기본 및 보조— 클러스터에는 동일한 수의 마스터 노드, 노드 유형, 엔진 버전 및 샤드 수(클러스터 모드가 활성화된 경우)가 있어야 합니다. 글로벌 데이터 스토어의 각 클러스터에는 해당 클러스터에 대한 로컬 읽기 트래픽을 수용하기 위해 다른 수의 읽기 복제본이 있을 수 있습니다.

기존 단일 노드 클러스터를 사용하려는 경우 복제를 활성화해야 합니다.

- 한 AWS 리전의 기본 클러스터에서 최대 두 개의 다른 AWS 리전의 보조 클러스터로 복제를 설정할 수 있습니다.
- VPC 클러스터에서만 글로벌 데이터 스토어로 작업할 수 있습니다. 자세한 내용은 [Amazon VPC에 있는 ElastiCache 클러스터에 액세스하기 위한 액세스 패턴 \(p. 174\)](#) 단원을 참조하십시오. EC2-Classic을 사용하는 경우 글로벌 데이터 스토어가 지원되지 않습니다.
- ElastiCache는 한 AWS 리전에서 다른 AWS 리전으로의 자동 장애 조치를 지원하지 않습니다. 필요한 경우 보조 클러스터를 수동으로 승격할 수 있습니다. 문제 해결 예는 [보조 클러스터를 기본 클러스터로 승격 \(p. 105\)](#)을(를) 참조하십시오.
- 기존 데이터에서 부트스트랩하려면 기존 클러스터를 기본 클러스터로 사용하여 글로벌 데이터 스토어를 생성합니다. 기존 클러스터를 보조 클러스터로 추가하는 것은 지원하지 않습니다. 클러스터를 보조 클러스터로 추가하는 프로세스는 데이터가 지워져 데이터가 손실될 수 있습니다.
- 파라미터 업데이트는 글로벌 데이터 스토어에 속한 클러스터의 로컬 파라미터 그룹을 수정할 때 모든 클러스터에 적용됩니다.
- 리전 클러스터를 수직(확장 및 축소) 및 수평(확장 및 축소)으로 확장할 수 있습니다. 글로벌 데이터 스토어를 수정하여 클러스터를 확장할 수 있습니다. 그러면 글로벌 데이터 스토어의 모든 리전 클러스터가 중단 없이 확장됩니다.

- 글로벌 데이터 스토어는 **고객 마스터 키**, **미사용 데이터 암호화**, **전송 중 데이터 암호화** 및 **Redis AUTH**를 지원합니다. 리전 간 통신에 대한 보안은 **VPC 피어링**을 통해 제공됩니다.

#### Note

글로벌 데이터 스토어는 다음 규정에 따라 **pub/sub 메시지**를 지원합니다.

- 클러스터 모드가 비활성화된 경우 pub/sub가 완전히 지원됩니다. 기본 AWS 리전의 마스터에 게시된 이벤트는 보조 AWS 리전으로 전파됩니다.
- 클러스터 모드가 활성화된 경우 다음 사항이 적용됩니다.
  - 키스페이스에 없는 게시된 이벤트의 경우 동일한 AWS 리전의 구독자만 이벤트를 수신합니다.
  - 게시된 키스페이스 이벤트의 경우 모든 AWS 리전의 구독자가 이벤트를 수신합니다.

## 글로벌 데이터 스토리지 사용(콘솔)

콘솔을 사용하여 글로벌 데이터 스토어를 생성하려면 다음 2단계 프로세스를 수행합니다.

1. 기존 클러스터를 사용하거나 새 클러스터를 생성하여 기본 클러스터를 생성합니다. 엔진은 Redis 5.0.6 이상이어야 합니다.
2. Redis 5.0.6 이상의 엔진을 사용하여 서로 다른 AWS 리전에 최대 2개의 보조 클러스터를 추가합니다.

다음 절차에서는 Redis용 글로벌 데이터 스토어를 생성하고 Redis용 ElastiCache 콘솔을 사용하여 다른 작업을 수행하는 방법에 대해 설명합니다.

#### 주제

- [기존 클러스터를 사용하여 글로벌 데이터 스토어 생성 \(p. 99\)](#)
- [새 기본 클러스터를 사용하여 새 글로벌 데이터 스토어 생성 \(p. 101\)](#)
- [글로벌 데이터 스토어 세부 정보 보기 \(p. 102\)](#)
- [글로벌 데이터 스토어에 리전 추가 \(p. 103\)](#)
- [글로벌 데이터 스토어 수정 \(p. 105\)](#)
- [보조 클러스터를 기본 클러스터로 승격 \(p. 105\)](#)
- [글로벌 데이터 스토어에서 리전 제거 \(p. 106\)](#)
- [글로벌 데이터 스토어 삭제 \(p. 106\)](#)

## 기존 클러스터를 사용하여 글로벌 데이터 스토어 생성

이 시나리오에서는 기존 클러스터를 사용하여 새 글로벌 데이터 스토어의 기본 클러스터 역할을 합니다. 그런 다음 별도의 AWS 리전에 보조 읽기 전용 클러스터를 생성합니다. 이 보조 클러스터는 기본 클러스터에서 자동 및 비동기 업데이트를 받습니다.

#### Important

기존 클러스터는 Redis 5.0.6 이상의 엔진을 사용해야 합니다.

기존 클러스터를 사용하여 글로벌 데이터 스토어를 생성하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 Redis를 선택한 후 클러스터를 선택합니다.
3. 작업에 대해 글로벌 데이터 스토어 설정을 선택합니다.
4. 글로벌 데이터 스토어 설정 페이지에서 다음을 수행합니다.

- 글로벌 데이터 스토어 이름 접미사 값 입력: 이 접미사는 글로벌 데이터 스토어의 고유 이름을 생성하는 데 사용됩니다. 여기에서 지정하는 접미사를 사용하여 글로벌 데이터 스토어를 검색할 수 있습니다.
  - (선택 사항) 설명 값을 입력합니다.
5. 보조 클러스터 세부 정보에서 클러스터가 저장될 다른 AWS 리전을 선택합니다.
  6. Redis 설정에서 클러스터의 이름에 대한 값과 필요에 따라 설명에 값을 입력합니다.
  7. 다음 옵션을 그대로 유지하십시오. 기본 클러스터 구성과 일치하도록 미리 채워져 있으므로 변경할 수 없습니다.

- 엔진 버전
- 노드 유형
- 파라미터 그룹

#### Note

ElastiCache는 제공된 파라미터 그룹의 값에서 새 파라미터 그룹을 자동으로 생성하고 새 파라미터 그룹을 클러스터에 적용합니다. 글로벌 데이터 스토어의 파라미터를 수정하려면 이 새 파라미터 그룹을 사용합니다. 자동 생성된 각 파라미터 그룹은 하나의 클러스터에만 연결되므로 하나의 글로벌 데이터 스토어에만 연결됩니다.

- 샤드 수
- 저장된 암호화

#### Note

고객 관리형 고객 마스터 키를 선택하고 키를 선택하여 다른 암호화 키를 제공할 수 있습니다.

- 전송 중 데이터 암호화
  - Redis AUTH
8. (선택 사항) 필요에 따라 나머지 보조 클러스터 설정을 업데이트합니다. 기본 클러스터와 동일한 값으로 미리 채워지지만 해당 클러스터에 대한 특정 요구 사항을 충족하도록 업데이트할 수 있습니다.

- 포트
- 복제본 개수
- Subnet Group
- 기본 가용 영역
- 보안 그룹
- 고객 관리형(고객 마스터 키)
- Redis AUTH 토큰
- 자동 백업 활성화
- 백업 보존 기간
- 백업 기간
- 유지 관리 기간
- SNS 알림에 대한 주제

9. 생성을 선택합니다. 이렇게 하면 글로벌 데이터 스토어의 상태가 생성 중으로 설정됩니다. 기본 클러스터가 글로벌 데이터 스토어와 연결되고 보조 클러스터가 연결 중 상태이면 상태가 수정 중으로 전환됩니다.

기본 클러스터 및 보조 클러스터가 글로벌 데이터 스토어와 연결되면 상태가 사용 가능으로 변경됩니다. 이 시점에서 읽기 및 쓰기를 허용하는 기본 클러스터와 기본 클러스터에서 복제된 읽기를 허용하는 보조 클러스터가 있습니다.

- 글로벌 데이터 스토어 – 클러스터가 속한 글로벌 데이터 스토어의 이름입니다.
- 글로벌 데이터 스토어 역할 – 클러스터의 역할(기본 또는 보조)입니다.

다른 AWS 리전에 보조 클러스터를 최대 1개까지 추가할 수 있습니다. 자세한 내용은 [글로벌 데이터 스토어에 리전 추가 \(p. 103\)](#) 단원을 참조하십시오.

## 새 기본 클러스터를 사용하여 새 글로벌 데이터 스토어 생성

새 글로벌 데이터 스토어를 생성하도록 선택한 경우 다음 절차를 따르십시오.

새 글로벌 데이터 스토어를 생성하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 글로벌 데이터 스토어를 선택한 다음 생성을 선택합니다.
3. 글로벌 데이터 스토어 생성에서 다음을 수행합니다.
  - a. 글로벌 데이터 스토어 이름 접미사를 값을 입력합니다. ElastiCache는 접미사를 사용하여 글로벌 데이터 스토어의 고유 이름을 생성합니다. 여기에서 지정하는 접미사를 사용하여 글로벌 데이터 스토어를 검색할 수 있습니다.
  - b. (선택 사항) 글로벌 데이터 스토어 설명에 값을 입력합니다.
4. 기본 클러스터 세부 정보에서 리전에 대해 사용 가능한 AWS 리전과 다음 옵션 중 하나를 선택합니다.
  - [새 리전 클러스터를 기본 클러스터로 생성 \(p. 101\)](#)를 선택하십시오.
  - [기존 클러스터를 기본 클러스터로 사용 \(p. 102\)](#)

## 새 리전 클러스터를 기본 클러스터로 생성

글로벌 데이터 스토어의 기본 클러스터로 사용할 새 리전 클러스터를 생성하려면 다음을 수행합니다.

새 리전 클러스터를 기본 클러스터로 생성하려면

1. 이름에 클러스터의 이름을 입력합니다.
2. (선택 사항) 설명에 클러스터에 대한 설명을 추가합니다.
3. 특정 요구 사항에 맞게 나머지 설정을 구성합니다. 파라미터 그룹을 선택하여 엔진 구성 값을 설정하면 해당 파라미터 그룹이 글로벌 데이터 스토어의 모든 클러스터에 적용됩니다. 파라미터 그룹 페이지에서 yes/no 글로벌 속성은 파라미터 그룹이 글로벌 데이터 스토어의 일부인지 여부를 나타냅니다.
4. [Next]를 선택합니다.
5. 보조 클러스터 세부 정보에서 클러스터가 저장될 다른 AWS 리전을 선택합니다.
6. Redis 설정에서 클러스터의 이름에 대한 값과 필요에 따라 설명에 값을 입력합니다.
7. 다음 필드는 기본 클러스터 구성과 일치하도록 미리 채워지며 변경할 수 없습니다.
  - 엔진 버전
  - 인스턴스 유형
  - 노드 유형
  - 샤드 수
  - 파라미터 그룹

### Note

ElastiCache는 제공된 파라미터 그룹의 값에서 새 파라미터 그룹을 자동으로 생성하고 새 파라미터 그룹을 클러스터에 적용합니다. 글로벌 데이터 스토어의 파라미터를 수정하려면 이

새 파라미터 그룹을 사용합니다. 자동 생성된 각 파라미터 그룹은 하나의 클러스터에만 연결되므로 하나의 글로벌 데이터 스토어에만 연결됩니다.

- 저장된 암호화

#### Note

고객 관리형 고객 마스터 키를 선택하고 키를 선택하여 다른 암호화 키를 제공하는 옵션이 있습니다.

- 전송 중 데이터 암호화
- Redis AUTH

나머지 보조 클러스터 설정은 기본 클러스터와 동일한 값으로 미리 채워지지만, 다음은 해당 클러스터에 대한 특정 요구 사항을 충족하도록 업데이트할 수 있습니다.

- 포트
  - 복제본 개수
  - Subnet Group
  - 기본 가용 영역
  - 보안 그룹
  - 고객 관리형(고객 마스터 키)
  - Redis AUTH 토큰
  - 자동 백업 활성화
  - 백업 보존 기간
  - 백업 기간
  - 유지 관리 기간
  - SNS 알림에 대한 주제
8. 생성을 선택합니다. 이렇게 하면 글로벌 데이터 스토어의 상태가 생성 중으로 설정됩니다. 기본 클러스터 및 보조 클러스터가 글로벌 데이터 스토어와 연결되면 상태가 사용 가능으로 변경됩니다. 읽기 및 쓰기 기능을 허용하는 기본 클러스터와 기본 클러스터에서 복제된 읽기를 허용하는 보조 클러스터가 있습니다.

또한 Redis 페이지가 업데이트되어 클러스터가 다음을 포함하여 글로벌 데이터 스토어의 일부인지 여부를 나타냅니다.

- 글로벌 데이터 스토어 – 클러스터가 속한 글로벌 데이터 스토어의 이름입니다.
- 글로벌 데이터 스토어 역할 – 클러스터의 역할(기본 또는 보조)을 반영합니다.

다른 AWS 리전에 보조 클러스터를 최대 1개까지 추가할 수 있습니다. 자세한 내용은 [글로벌 데이터 스토어에 리전 추가 \(p. 103\)](#) 단원을 참조하십시오.

## 기존 클러스터를 기본 클러스터로 사용

이 옵션을 선택한 경우 5단계부터 시작하는 [기존 클러스터를 사용하여 글로벌 데이터 스토어 생성 \(p. 99\)](#)의 단계를 따릅니다.

## 글로벌 데이터 스토어 세부 정보 보기

기존 글로벌 데이터 스토어의 세부 정보를 보고 글로벌 데이터 스토어 페이지에서 수정할 수도 있습니다.

글로벌 데이터 스토어 세부 정보를 보려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 글로벌 데이터 스토어를 선택한 다음 사용 가능한 글로벌 데이터 스토어를 선택합니다.

그러면 다음 글로벌 데이터 스토어 속성을 검사할 수 있습니다.

- 글로벌 데이터 스토어 이름: 글로벌 데이터 스토어의 이름입니다.
- 설명: 글로벌 데이터 스토어에 대한 설명입니다.
- 상태: 옵션은 다음과 같습니다.
  - 생성 중
  - [Modifying]
  - 응시 가능
  - 삭제 중
- 기본 전용 - 이 상태는 글로벌 데이터 스토어에 기본 클러스터만 포함되어 있음을 나타냅니다. 모든 보조 클러스터가 삭제되거나 성공적으로 생성되지 않습니다.
- 클러스터 모드: 활성화되거나 비활성화됩니다.
- Redis 엔진 버전: 글로벌 데이터 스토어를 실행하는 Redis 엔진 버전입니다.
- 인스턴스 노드 유형: 글로벌 데이터 스토어에 사용되는 노드 유형입니다.
- 미사용 데이터 암호화: 활성화되거나 비활성화됩니다.
- 전송 중 데이터 암호화: 활성화되거나 비활성화됩니다.
- Redis AUTH: 활성화되거나 비활성화됩니다.

글로벌 데이터 스토어를 다음과 같이 변경할 수 있습니다.

- [글로벌 데이터 스토어에 리전 추가 \(p. 103\)](#)
- [글로벌 데이터 스토어에서 리전 제거 \(p. 106\)](#)
- [보조 클러스터를 기본 클러스터로 승격 \(p. 105\)](#)
- [글로벌 데이터 스토어 수정 \(p. 105\)](#)

글로벌 데이터 스토어 페이지에는 글로벌 데이터 스토어를 구성하는 개별 클러스터와 각각에 해당하는 다음 속성도 나열됩니다.

- 리전 - 클러스터가 저장된 AWS 리전입니다.
- 역할 - 기본 또는 보조입니다.
- 클러스터 이름 - 클러스터의 이름입니다.
- 상태 - 옵션은 다음과 같습니다.
  - 연결 중 - 클러스터가 글로벌 데이터 스토어에 연결되는 중입니다.
  - 연관됨 - 클러스터가 글로벌 데이터 스토어에 연결되어 있습니다.
  - 연결 해제 중 - 글로벌 데이터 스토어 이름을 사용하여 글로벌 데이터 스토어에서 보조 클러스터를 제거하는 중입니다. 이후에는 보조 클러스터가 더 이상 기본 클러스터에서 업데이트를 수신하지 않지만 해당 AWS 리전에서 독립 실행형 클러스터로 유지됩니다.
  - 연결 해제됨 - 보조 클러스터가 글로벌 데이터 스토어에서 제거되었으며 이제 AWS 리전에서 독립 실행형 클러스터가 되었습니다.
- 글로벌 데이터 스토어 지연 - 글로벌 데이터 스토어에 보조 AWS 리전당 하나의 값을 표시합니다. 보조 리전의 마스터 노드와 기본 리전의 마스터 노드 간의 지연입니다. 클러스터 모드가 활성화된 Redis의 경우 지연은 샤드 간의 최대 지연을 나타냅니다.

## 글로벌 데이터 스토어에 리전 추가

기존 글로벌 데이터 스토어에 최대 하나의 AWS 리전을 추가할 수 있습니다. 이 시나리오에서는 기본 클러스터로부터 자동 및 비동기 업데이트를 수신하는 별도의 AWS 리전에 읽기 전용 클러스터를 생성합니다.

## 글로벌 데이터 스토어에 AWS 리전을 추가하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 글로벌 데이터 스토어를 선택한 다음 글로벌 데이터 스토어 이름에서 글로벌 데이터 스토어를 선택합니다.
3. 리전 추가를 선택합니다.
4. 보조 클러스터가 상주할 AWS 리전을 선택합니다.
5. Redis 설정에서 클러스터의 이름에 대한 값과 필요에 따라 설명에 값을 입력합니다.
6. 다음 옵션을 그대로 유지하십시오. 기본 클러스터 구성과 일치하도록 미리 채워져 있으므로 변경할 수 없습니다.

- 엔진 버전
- 인스턴스 유형
- 노드 유형
- 샤드 수
- 파라미터 그룹

### Note

ElastiCache는 제공된 파라미터 그룹의 값에서 새 파라미터 그룹을 자동으로 생성하고 새 파라미터 그룹을 클러스터에 적용합니다. 글로벌 데이터 스토어의 파라미터를 수정하려면 이 새 파라미터 그룹을 사용합니다. 자동 생성된 각 파라미터 그룹은 하나의 클러스터에만 연결되므로 하나의 글로벌 데이터 스토어에만 연결됩니다.

- 저장된 암호화

### Note

고객 관리형 고객 마스터 키를 선택하고 키를 선택하여 다른 암호화 키를 제공하는 옵션이 있습니다.

- 전송 중 데이터 암호화
  - Redis AUTH
7. (선택 사항) 필요에 따라 나머지 보조 클러스터 설정을 업데이트합니다. 기본 클러스터와 동일한 값으로 미리 채워지지만 해당 클러스터에 대한 특정 요구 사항을 충족하도록 업데이트할 수 있습니다.

- 포트
- 복제본 개수
- Subnet Group
- 기본 가용 영역
- 보안 그룹
- 고객 관리형(고객 마스터 키)
- Redis AUTH 토큰
- 자동 백업 활성화
- 백업 보존 기간
- 백업 기간
- 유지 관리 기간
- SNS 알림에 대한 주제

8. 추가를 선택합니다.

## 글로벌 데이터 스토어 수정

리전 클러스터의 속성을 수정할 수 있습니다. 보조 클러스터를 기본 클러스터로 승격하는 경우를 제외하고 글로벌 데이터 스토어에서는 하나의 수정 작업만 진행 중일 수 있습니다. 자세한 내용은 [보조 클러스터를 기본 클러스터로 승격 \(p. 105\)](#) 단원을 참조하십시오.

글로벌 데이터 스토어를 수정하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 글로벌 데이터 스토어를 선택한 다음 글로벌 데이터 스토어 이름에 글로벌 데이터 스토어를 선택합니다.
3. 수정을 선택하고 다음 옵션 중에서 선택합니다.
  - 설명 수정 – 글로벌 데이터 스토어에 대한 설명을 업데이트합니다.
  - 엔진 버전 수정 – Redis 엔진 버전 5.0.6 이상만 사용할 수 있습니다.
  - 노드 유형 수정 – 리전 클러스터를 수직(확장 및 축소) 및 수평(확장 및 축소)으로 확장합니다. 옵션에는 R5 및 M5 노드 패밀리가 포함됩니다. 노드 유형에 대한 자세한 내용은 [지원되는 노드 유형 \(p. 36\)](#) 단원을 참조하십시오.
  - 자동 장애 조치 수정 – 장애 조치를 활성화하거나 비활성화합니다. 장애 조치를 활성화하고 리전 클러스터의 기본 노드가 죽으면 ElastiCache는 리전 복제본 중 하나로 장애 조치를 합니다. 자세한 내용은 [자동 장애 조치](#)를 참조하십시오.

클러스터 모드가 활성화된 Redis 클러스터의 경우:

- 샤드 추가 – 추가할 샤드 수를 입력하고 선택적으로 하나 이상의 가용 영역을 지정합니다.
- 샤드 삭제 – 각 AWS 리전에서 삭제할 샤드를 선택합니다.
- 샤드 재분배 – 슬롯 분포를 재분배하여 클러스터의 기존 샤드 간에 균일한 분포를 보장합니다.

글로벌 데이터 스토어의 파라미터를 수정하려면 글로벌 데이터 스토어에 대한 멤버 클러스터의 파라미터 그룹을 수정합니다. ElastiCache는 이 변경 사항을 해당 글로벌 데이터 스토어 내의 모든 클러스터에 적용합니다. 해당 클러스터의 파라미터 그룹을 수정하려면 Redis 콘솔 또는 [ModicCachCluster](#) API를 사용합니다. 자세한 내용은 [파라미터 그룹 수정 \(p. 124\)](#) 단원을 참조하십시오. 글로벌 데이터 스토어 내에 포함된 클러스터의 파라미터 그룹을 수정하면 해당 글로벌 데이터 스토어 내의 모든 클러스터에 적용됩니다.

전체 파라미터 그룹 또는 특정 파라미터 재설정하려면 [ResetCacheParameterGroup](#) API를 사용합니다.

## 보조 클러스터를 기본 클러스터로 승격

성능 문제가 발생하여 기본 클러스터 또는 AWS 리전을 사용할 수 없게 되면 보조 클러스터를 기본 클러스터로 승격할 수 있습니다. 다른 수정이 진행 중이더라도 언제든지 승격이 허용됩니다. 또한 여러 승격을 병렬로 실행할 수 있으며 글로벌 데이터 스토어가 최종적으로 하나의 기본 클러스터가 됩니다. 여러 보조 클러스터를 동시에 승격하는 경우 Redis용 ElastiCache에서 궁극적으로 하나의 클러스터가 기본 클러스터가 되는 것을 보장하지 않습니다.

보조 클러스터를 기본 클러스터로 승격하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 Redis 아래 글로벌 데이터 스토어를 선택합니다.
3. 세부 정보를 보려면 글로벌 데이터 스토어 이름을 선택합니다.
4. 보조 클러스터 선택
5. 기본 클러스터로 승격을 선택합니다.

그러면 Promoting a region to primary will make the cluster in this region as read/writable. Are you sure you want to promote the **secondary** cluster to primary? 경고와 함께 결정을 확인하라는 메시지가 표시됩니다.

The current primary cluster in **primary region** will become secondary and will stop accepting writes after this operation completes. Please ensure you update your application stack to direct traffic to the new primary region.

6. 승력을 계속하려면 확인을 선택하고 그렇지 않으면 취소를 선택합니다.

확인하려면 글로벌 데이터 스토어가 수정 중 상태로 전환되어 승격이 완료될 때까지 사용할 수 없습니다.

## 글로벌 데이터 스토어에서 리전 제거

다음 절차를 사용하여 글로벌 데이터 스토어에서 AWS 리전을 제거할 수 있습니다.

글로벌 데이터 스토어에서 AWS 리전을 제거하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 Redis 아래 글로벌 데이터 스토어를 선택합니다.
3. 글로벌 데이터 스토어를 선택합니다.
4. 제거할 리전을 선택합니다.
5. 리전 제거를 선택합니다.

### Note

이 옵션은 보조 클러스터에만 사용할 수 있습니다.

그러면 Removing the region will remove your only available cross region replica for the primary cluster. Your primary cluster will no longer be set up for disaster recovery and improved read latency in remote region. Are you sure you want to remove the selected region from the global datastore? 경고와 함께 결정을 확인하라는 메시지가 표시됩니다.

6. 승력을 계속하려면 확인을 선택하고 그렇지 않으면 취소를 선택합니다.

확인을 선택하면 AWS 리전이 제거되고 보조 클러스터는 더 이상 복제 업데이트를 수신하지 않습니다.

## 글로벌 데이터 스토어 삭제

글로벌 데이터 스토어를 삭제하려면 먼저 모든 보조 클러스터를 제거합니다. 자세한 내용은 [글로벌 데이터 스토어에서 리전 제거 \(p. 106\)](#) 단원을 참조하십시오. 이렇게 하면 글로벌 데이터 스토어가 기본 전용 상태로 유지됩니다.

글로벌 데이터 스토어를 삭제하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 Redis 아래 글로벌 데이터 스토어를 선택합니다.
3. 글로벌 데이터 스토어 이름에서 삭제할 글로벌 데이터 스토어를 선택한 다음 삭제를 선택합니다.

그러면 Are you sure you want to delete this Global Datastore? 경고와 함께 결정을 확인하라는 메시지가 표시됩니다.

4. 삭제를 선택합니다.

글로벌 데이터 스토어가 삭제 중 상태로 전환됩니다.

## 글로벌 데이터 저장소(CLI) 사용

AWS CLI(AWS Command Line Interface)를 사용하면 명령줄에서 여러 AWS 제품을 관리하고 스크립트를 통해 자동화할 수 있습니다. 임시(일회성) 작업에 AWS CLI를 사용할 수 있습니다.

### AWS CLI 다운로드 및 구성

AWS CLI는 Windows, macOS 또는 Linux에서 실행됩니다. 다음 절차에 따라 다운로드 및 구성합니다.

CLI를 다운로드, 설치 및 구성하려면

1. [AWS 명령줄 인터페이스](#) 웹 페이지에서 AWS CLI를 다운로드합니다.
2. AWS Command Line Interface 사용 설명서의 AWS CLI 설치 및 AWS CLI 구성 지침을 따릅니다.

### 글로벌 데이터 스토어에 AWS CLI 사용

글로벌 데이터 스토어를 사용하려면 다음 CLI 작업을 사용합니다.

- [create-global-replication-group](#)

```
aws elasticache create-global-replication-group \
 --global-replication-group-id-suffix my global datastore \
 --primary-replication-group-id sample-repl-group \
 --global-replication-group-description an optional description of the global
 datastore
```

- [create-replication-group](#) – 이 작업을 사용하여 글로벌 데이터 스토어의 이름을 --global-replication-group-id 파라미터에 제공하여 글로벌 데이터 스토어에 대한 보조 클러스터를 생성합니다.

```
aws elasticache create-replication-group \
 --replication-group-id secondary replication group name \
 --replication-group-description "Replication group description" \
 --global-replication-group-id global datastore name \
 --primary-replication-group-id sample-repl-group
```

- [describe-global-replication-groups](#)

```
aws elasticache describe-global-replication-groups \
 --global-replication-group-id my global datastore \
 --show-member-info an optional parameter that returns a list of the primary and
 secondary clusters that make up the global datastore
```

- [modify-global-replication-group](#)

```
aws elasticache modify-global-replication-group \
 --global-replication-group-id my global datastore \
 --automatic-failover-enabled yes/no \
 --cache-node-type node type \
 --engine-version engine version \
 --apply-immediately \
 --global-replication-group-description description
```

- [delete-global-replication-group](#)

```
aws elasticache delete-global-replication-group \
 --global-replication-group-id my global datastore \
 --retain-primary-replication-group defaults to true
```

- [disassociate-global-replication-group](#)

```
aws elasticache disassociate-global-replication-group \
 --global-replication-group-id my Global Datastore \
 --replication-group-id my secondary cluster \
 --replication-group-region the AWS Region in which the secondary cluster resides
```

- [failover-global-replication-group](#)

```
aws elasticache failover-replication-group \
 --global-replication-group-id my global datastore \
 --primary-region The AWS Region of the primary cluster \
 --primary-replication-group-id The name of the global datastore, including the
 suffix.
```

- [increase-node-groups-in-global-replication-group](#)

```
aws elasticache increase-node-groups-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name \
 --node-group-count 3
```

- [decrease-node-groups-in-global-replication-group](#)

```
aws elasticache decrease-node-groups-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name \
 --node-group-count 3
```

- [rebalance-shards-in-global-replication-group](#)

```
aws elasticache rebalance-shards-in-global-replication-group \
 --apply-immediately yes \
 --global-replication-group-id global-replication-group-name \
 --node-group-count 3
```

다음말을 사용하여 사용 가능한 모든 명령 Redis용 ElastiCache을 나열합니다.

```
aws elasticache help
```

다음말을 사용하면 특정 명령을 설명하고 그 사용법에 대해 자세히 알아볼 수도 있습니다.

```
aws elasticache create-global-replication-group help
```

## 연결 엔드포인트 찾기

애플리케이션에서 엔드포인트를 사용하여 클러스터에 연결합니다. 엔드포인트는 노드나 클러스터의 고유한 주소입니다.

## 사용할 엔드포인트

- Memcached 클러스터 - 자동 검색을 사용하는 경우 클러스터의 구성 엔드포인트를 사용하여 Memcached 클라이언트를 구성할 수 있으므로 자동 검색을 지원하는 클라이언트를 사용해야 합니다.

자동 검색을 사용하지 않으면 읽기 및 쓰기를 위해 개별 노드 엔드포인트를 사용하도록 클라이언트를 구성해야 합니다. 또한 노드를 추가 및 제거할 때 엔드포인트를 추적해야 합니다.

다음 단원에서는 실행 중인 엔진에 필요한 엔드포인트를 찾는 방법을 안내합니다.

### 주제

- [클러스터의 엔드포인트 찾기 \(콘솔\) \(p. 109\)](#)
- [엔드포인트 찾기 \(AWS CLI\) \(p. 110\)](#)
- [엔드포인트 찾기 \(ElastiCache API\) \(p. 112\)](#)

## 클러스터의 엔드포인트 찾기 (콘솔)

모든 Memcached 엔드포인트는 읽기/쓰기 엔드포인트입니다. Memcached 클러스터에 있는 노드에 연결하기 위해 애플리케이션은 각 노드의 엔드포인트를 사용하거나 자동 검색과 함께 클러스터의 구성 엔드포인트를 사용할 수 있습니다. 자동 검색을 사용하려면 자동 검색을 지원하는 클라이언트를 사용해야 합니다.

자동 검색을 사용하면 클라이언트 애플리케이션이 구성 엔드포인트를 사용하여 Memcached 클러스터에 연결합니다. 노드를 추가하거나 제거하여 클러스터를 조정할 때는 애플리케이션에서 클러스터의 모든 노드를 자동으로 "인식"하고 그 중에서 어디에나 연결할 수 있습니다. 자동 검색을 사용하지 않으면 애플리케이션에서 이 작업을 수행하거나, 사용자가 노드를 추가하거나 제거할 때마다 애플리케이션의 엔드포인트를 수동으로 업데이트해야 합니다. 자동 검색에 대한 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오.

다음 절차에서는 ElastiCache 콘솔을 사용하여 클러스터의 구성 엔드포인트 복사본을 찾거나 노드 엔드포인트를 찾는 방법을 설명합니다.

### Memcached 클러스터의 엔드포인트를 찾거나 복사하려면(console)

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 [Memcached]를 선택합니다.

캐시 클러스터 화면에 Memcached 클러스터 목록이 나타납니다.

3. 엔드포인트의 Memcached 클러스터를 찾습니다.

구성 엔드포인트만 필요한 경우 작업이 완료되었습니다. 구성 엔드포인트는 [Configuration Endpoint] 옆에 있으며 `clusterName.xxxxxx.cfg.usw2.cache.amazonaws.com:port`과 같이 표시됩니다.

개별 노드 엔드포인트를 표시하거나 클립보드에 엔드포인트를 복사하려면 [Copy Node Endpoint]를 선택합니다.

## Copy Node Endpoint

### Configuration Endpoint

`.cfg.usw2.cache.amazonaws.com:11211`

Use the ElastiCache Cluster Client and Configuration Endpoint to automatically discover hosts.

[Download](#) the client.

### Node Endpoints

`.0001.usw2.cache.amazonaws.com:11211`

`.0002.usw2.cache.amazonaws.com:11211`

`.0003.usw2.cache.amazonaws.com:11211`

Memcached 클러스터의 엔드포인트

4. 엔드포인트를 클립보드에 복사하려면
  - a. [Copy Node Endpoint] 화면에서 복사할 엔드포인트를 강조 표시합니다.
  - b. 강조 표시한 엔드포인트를 마우스 오른쪽 버튼으로 클릭한 후 컨텍스트 메뉴에서 복사를 선택합니다.

강조 표시한 엔드포인트가 클립보드에 복사됩니다.

구성 엔드포인트와 노드 엔드포인트는 매우 비슷합니다. 두 엔드포인트의 차이는 다음과 같이 굵게 표시됩니다.

```
myclustername.*****.cfg.usw2.cache.amazonaws.com:port # configuration endpoint contains "cfg"
myclustername.*****.0001.usw2.cache.amazonaws.com:port # node endpoint for node 0001
```

### Important

Memcached 구성 엔드포인트의 CNAME을 만들기로 선택하면 CNAME에 `.cfg`를 포함해야 자동 검색 클라이언트가 CNAME을 구성 엔드포인트로 인식합니다.

## 엔드포인트 찾기 (AWS CLI)

Amazon ElastiCache용 AWS CLI를 사용하여 노드 및 클러스터의 엔드포인트를 찾을 수 있습니다.

주제

- [노드 및 클러스터의 엔드포인트 찾기 \(AWS CLI\) \(p. 110\)](#)

## 노드 및 클러스터의 엔드포인트 찾기 (AWS CLI)

AWS CLI 사용하여 `describe-cache-clusters` 명령으로 클러스터 및 해당 노드의 엔드포인트를 찾을 수 있습니다. Memcached 클러스터의 경우 명령이 구성 엔드포인트를 반환합니다. 또한 선택적 파라미터 `--show-cache-node-info`를 포함할 경우 명령이 클러스터에 있는 개별 노드의 엔드포인트를 반환합니다.

### Example

다음 명령은 Memcached 클러스터 `mycluster`의 구성 엔드포인트(ConfigurationEndpoint) 및 개별 노드 엔드포인트(Endpoint)를 검색합니다.

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
엔드포인트 찾기 (AWS CLI)

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

위 작업의 출력은 다음과 같습니다(JSON 형식).

```
{
 "CacheClusters": [
 {
 "Engine": "memcached",
 "CacheNodes": [
 {
 "CacheNodeId": "0001",
 "Endpoint": {
 "Port": 11211,
 "Address": "mycluster.1abc4d.0001.usw2.cache.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.1abc4d.0002.usw2.cache.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.1abc4d.0003.usw2.cache.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.1abc4d.cfg.usw2.cache.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

Memcached 구성 엔드포인트의 CNAME을 만들기로 선택하면 CNAME에 .cfg.를 포함해야 PHP 클라이언트가 CNAME을 구성 엔드포인트로 인식합니다. 예를 들어, mycluster.cfg.local가 session.save\_path 파라미터의 php.ini 파일에 있습니다.

자세한 내용은 [describe-cache-clusters](#) 단원을 참조하십시오.

## 엔드포인트 찾기 (ElastiCache API)

Amazon ElastiCache API를 사용하여 노드 및 클러스터의 엔드포인트를 찾을 수 있습니다.

### 주제

- [노드 및 클러스터의 엔드포인트 찾기 \(ElastiCache API\)](#) (p. 112)

## 노드 및 클러스터의 엔드포인트 찾기 (ElastiCache API)

ElastiCache API를 사용하여 DescribeCacheClusters 작업으로 클러스터 및 해당 노드의 엔드포인트를 찾을 수 있습니다. Memcached 클러스터의 경우 명령이 구성 엔드포인트를 반환합니다. 또한 선택적 파라미터 ShowCacheNodeInfo를 포함할 경우 작업이 클러스터에 있는 개별 노드의 엔드포인트를 반환합니다.

### Example

다음 명령은 Memcached 클러스터 mycluster의 구성 엔드포인트(ConfigurationEndpoint) 및 개별 노드 엔드포인트(Endpoint)를 검색합니다.

```
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

Memcached 구성 엔드포인트의 CNAME을 만들기로 선택하면 CNAME에 .cfg.를 포함해야 PHP 클라이언트가 CNAME을 구성 엔드포인트로 인식합니다. 예를 들어, mycluster.cfg.local가 session.save\_path 파라미터의 php.ini 파일에 있습니다.

## ElastiCache for Memcached 클러스터 조정

애플리케이션에서 처리해야 하는 데이터의 양은 거의 정적이 아닙니다. 비즈니스가 성장하거나 수요에서 일 반적인 변동을 경험할 경우 데이터의 양이 증가하거나 감소합니다. 캐시를 자체적으로 관리할 경우 최고의 수요에 대해 충분한 하드웨어를 프로비저닝해야 하므로, 비용이 많이 들 수 있습니다. Amazon ElastiCache 를 사용하면 현재 수요에 맞게 조정할 수 있어, 사용한 만큼만 요금을 지불할 수 있습니다. ElastiCache를 통 해 수요에 맞게 캐시를 조정할 수 있습니다.

다음은 수행하려는 조정 작업에 대한 올바른 주제를 찾는 데 도움이 됩니다.

### Memcached 클러스터 조정

| 작업       | 주제/링크                                      |
|----------|--------------------------------------------|
| 확장       | <a href="#">클러스터에 노드 추가 (p. 79)</a>        |
| 축소       | <a href="#">클러스터에서 노드 제거 (p. 83)</a>       |
| 노드 유형 변경 | <a href="#">수직으로 Memcached 조정 (p. 114)</a> |

Memcached 클러스터는 1~20개의 노드로 구성되어 있습니다. Memcached 클러스터를 스케일 아웃하고 스 케일 인하는 것은 클러스터에서 노드를 추가 또는 제거하는 것만큼 쉽습니다.

Memcached 클러스터에서 20개를 초과하는 노드가 필요할 경우 또는 AWS 리전에서 총 100개를 초과하는 노드가 필요할 경우 <https://aws.amazon.com/contact-us/elasticache-node-limit-request/>에서 ElastiCache 한 도 증가 요청 양식을 작성하십시오.

Memcached 클러스터의 모든 노드로 데이터를 분할할 수 있으므로 더 많은 양의 메모리가 있는 노드 유형으 로 스케일 업할 필요가 거의 없습니다. 그러나 Memcached 엔진이 데이터를 유지하지 않으므로 다른 노드 유 형으로 조정할 경우 애플리케이션에서 채우지 않으면 새 클러스터가 비워집니다.

### 주제

- [수평으로 Memcached 조정 \(p. 113\)](#)
- [수직으로 Memcached 조정 \(p. 114\)](#)

## 수평으로 Memcached 조정

Memcached 엔진은 여러 노드에 대한 데이터 분할을 지원합니다. 따라서 Memcached 클러스터는 쉽게 수평 으로 조정됩니다. Memcached 클러스터에는 1 ~ 20개의 노드가 있을 수 있습니다. 수평으로 Memcached 클 러스터를 조정하려면 노드를 추가 또는 제거하면 됩니다.

Memcached 클러스터에서 20개를 초과하는 노드가 필요할 경우 또는 AWS 리전에서 총 100개를 초과하는 노드가 필요할 경우 <https://aws.amazon.com/contact-us/elasticache-node-limit-request/>에서 ElastiCache 한 도 증가 요청 양식을 작성하십시오.

다음 항목은 노드를 추가 또는 제거하여 Memcached 클러스터를 스케일 아웃 또는 스케일 인하는 방법을 자 세히 설명합니다.

- [클러스터에 노드 추가 \(p. 79\)](#)
- [클러스터에서 노드 제거 \(p. 83\)](#)

Memcached 클러스터의 노드 수를 변경할 때마다 키스페이스의 일부라도 다시 매핑해야 정확한 노드에 매핑됩니다. Memcached 클러스터의 로드 밸런싱에 대한 자세한 내용은 [효율적인 로드 밸런싱을 위해 ElastiCache 클라이언트 구성 \(p. 25\)](#) 단원을 참조하십시오.

Memcached 클러스터에 대해 Auto Discovery를 사용할 경우 노드를 추가 또는 제거할 때 애플리케이션에 있는 엔드포인트를 변경할 필요가 없습니다. 자동 검색에 대한 자세한 내용은 [Memcached 클러스터의 노드를 자동으로 식별 \(p. 46\)](#) 단원을 참조하십시오. Auto Discovery를 사용하지 않을 경우 Memcached 클러스터의 노드 수를 변경할 때마다 애플리케이션에 있는 엔드포인트를 업데이트해야 합니다.

## 수직으로 Memcached 조정

Memcached 클러스터를 위 또는 아래로 확장하거나 축소하는 경우 새 클러스터를 생성해야 합니다. Memcached 클러스터는 애플리케이션이 채울 때까지 항상 비어 있습니다.

### Important

소형 노드 유형으로 스케일 다운할 경우 소형 노드 유형이 데이터 및 오버헤드에 적합해야 합니다. 자세한 내용은 [Memcached 노드 크기 선택 \(p. 68\)](#) 단원을 참조하십시오.

### 주제

- [수직으로 Memcached 조정 \(콘솔\) \(p. 114\)](#)
- [수직으로 Memcached 조정 \(AWS CLI\) \(p. 114\)](#)
- [수직으로 Memcached 조정 \(ElastiCache API\) \(p. 114\)](#)

## 수직으로 Memcached 조정 (콘솔)

다음 절차는 ElastiCache 콘솔을 사용하여 클러스터를 수직으로 조정하는 방법을 안내합니다.

Memcached 클러스터를 수직으로 조정하려면 (console)

1. 새 노드 유형으로 새 클러스터를 생성합니다. 자세한 내용은 [Memcached 클러스터 생성 \(콘솔\) \(p. 69\)](#) 단원을 참조하십시오.
2. 애플리케이션에서 엔드포인트를 새 클러스터의 엔드포인트로 업데이트합니다. 자세한 내용은 [클러스터의 엔드포인트 찾기 \(콘솔\) \(p. 109\)](#) 단원을 참조하십시오.
3. 이전 클러스터를 삭제합니다. 자세한 내용은 [AWS Management 콘솔 사용 \(p. 88\)](#) 단원을 참조하십시오.

## 수직으로 Memcached 조정 (AWS CLI)

다음 절차는 AWS CLI를 사용하여 Memcached 캐시 클러스터를 수직으로 조정하는 방법을 안내합니다.

Memcached 캐시 클러스터를 수직으로 조정하려면 (AWS CLI)

1. 새 노드 유형으로 새 캐시 클러스터를 생성합니다. 자세한 내용은 [클러스터 생성 \(AWS CLI\) \(p. 71\)](#) 단원을 참조하십시오.
2. 애플리케이션에서 엔드포인트를 새 클러스터의 엔드포인트로 업데이트합니다. 자세한 내용은 [엔드포인트 찾기 \(AWS CLI\) \(p. 110\)](#) 단원을 참조하십시오.
3. 이전 캐시 클러스터를 삭제합니다. 자세한 내용은 [AWS CLI 사용 \(p. 89\)](#) 단원을 참조하십시오.

## 수직으로 Memcached 조정 (ElastiCache API)

다음 절차는 ElastiCache API를 사용하여 Memcached 캐시 클러스터를 수직으로 조정하는 방법을 안내합니다.

Memcached 캐시 클러스터를 수직으로 조정하려면 (ElastiCache API)

1. 새 노드 유형으로 새 캐시 클러스터를 생성합니다. 자세한 내용은 [클러스터 생성 \(ElastiCache API\) \(p. 72\)](#) 단원을 참조하십시오.

2. 애플리케이션에서 엔드포인트를 새 캐시 클러스터의 엔드포인트로 업데이트합니다. 자세한 내용은 [엔드포인트 찾기 \(ElastiCache API\) \(p. 112\)](#) 단원을 참조하십시오.
3. 이전 캐시 클러스터를 삭제합니다. 자세한 내용은 [ElastiCache API 사용 \(p. 89\)](#) 단원을 참조하십시오.

## 파라미터 그룹을 사용해 엔진 파라미터 구성

Amazon ElastiCache는 파라미터를 사용하여 노드 및 클러스터의 런타임 속성을 제어합니다. 일반적으로 최신 엔진 버전에는 새로운 기능을 지원하는 추가 파라미터가 포함됩니다. 파라미터가 정리된 표는 [Memcached 특정 파라미터 \(p. 127\)](#) 단원을 참조하십시오.

max\_cache\_memory와 같은 일부 파라미터 값은 엔진 및 노드 유형에 의해 결정됩니다. 노드 유형별 파라미터 값의 표는 [Memcached 노드 유형별 파라미터 \(p. 137\)](#) 단원을 참조하십시오.

### Note

Memcached 특정 파라미터 목록은 [Memcached 특정 파라미터](#)를 참조하십시오.

### 주제

- [파라미터 관리 \(p. 115\)](#)
- [캐시 파라미터 그룹 티어 \(p. 116\)](#)
- [파라미터 그룹 생성 \(p. 116\)](#)
- [이름별로 파라미터 그룹 목록 조회 \(p. 119\)](#)
- [파라미터 그룹의 값 목록 조회 \(p. 123\)](#)
- [파라미터 그룹 수정 \(p. 124\)](#)
- [파라미터 그룹 삭제 \(p. 126\)](#)
- [Memcached 특정 파라미터 \(p. 127\)](#)

## 파라미터 관리

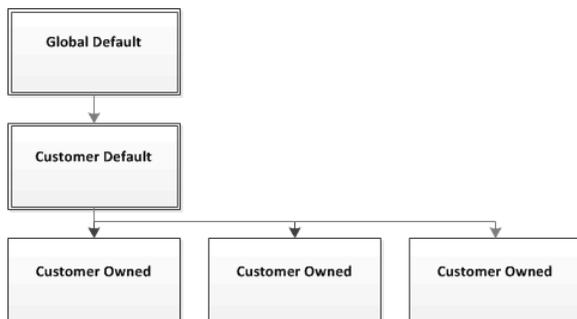
더욱 쉬운 파라미터 관리를 위해 파라미터를 명명된 파라미터 그룹으로 그룹화합니다. 파라미터 그룹은 시작하는 동안 엔진 소프트웨어에 전달되는 파라미터의 특정 값 조합을 나타냅니다. 이 값은 각 노드의 엔진 프로세서가 런타임에 작동하는 방식을 결정합니다. 특정 파라미터 그룹의 파라미터 값은 해당 파라미터가 속한 클러스터와 상관없이 그룹과 연결된 모든 노드에 적용됩니다.

클러스터 성능을 미세 조정하려면 일부 파라미터 값을 수정하거나 클러스터의 파라미터 그룹을 변경할 수 있습니다.

- 기본 파라미터 그룹을 수정하거나 삭제할 수 없습니다. 사용자 지정 파라미터 값이 필요하다면 사용자 지정 파라미터 그룹을 생성해야 합니다.
- 파라미터 그룹 패밀리와 할당할 클러스터는 호환 가능해야 합니다. 예를 들어 클러스터에서 Memcached 버전 1.4.8을 실행 중이라면 Memcached 1.4 패밀리의 기본 또는 사용자 지정 파라미터 그룹만 사용할 수 있습니다.
- 클러스터의 파라미터 그룹을 변경하면 조건부로 수정 가능한 파라미터의 값이 현재 및 새 파라미터 그룹에서 동일해야 합니다.
- 클러스터의 파라미터를 변경하면 변경 사항은 또는 클러스터가 재시작된 즉시 또는 그 이후에 클러스터에 적용됩니다. 이는 클러스터의 파라미터 그룹 자체에서 변경하는 파라미터 값을 클러스터의 파라미터 그룹 내에서 변경하는 마찬가지입니다. 특정 파라미터의 변경 사항이 적용되는 시점을 확인하려면 [Memcached 특정 파라미터 \(p. 127\)](#)에 대한 표에서 변경 사항 적용 열을 참조하십시오. 클러스터 재부팅에 관한 자세한 내용은 [클러스터 재부팅 \(p. 77\)](#) 단원을 참조하십시오.

## 캐시 파라미터 그룹 티어

Amazon ElastiCache에는 다음과 같이 세 가지 캐시 파라미터 그룹 티어가 있습니다.



### Amazon ElastiCache 파라미터 그룹 티어

#### 전역 기본값

해당 리전의 모든 Amazon ElastiCache 고객을 위한 최상위 루트 파라미터 그룹입니다.

전역 기본 캐시 파라미터 그룹:

- ElastiCache용으로 예약되어 있으며 고객이 사용할 수 없습니다.

#### 고객 기본값

고객의 사용을 위해 생성된 전역 기본 캐시 파라미터 그룹의 사본입니다.

고객 기본 캐시 파라미터 그룹:

- ElastiCache가 생성하고 소유합니다.
- 고객이 이 캐시 파라미터 그룹에서 지원하는 엔진 버전을 실행 중인 모든 클러스터의 캐시 파라미터 그룹으로 사용할 수 있습니다.
- 고객이 편집할 수 없습니다.

#### 고객 소유

고객 기본 캐시 파라미터 그룹의 사본입니다. 고객 소유 캐시 파라미터 그룹은 고객이 캐시 파라미터 그룹을 생성할 때마다 만들어집니다.

고객 소유 캐시 파라미터 그룹:

- 고객이 생성하고 소유합니다.
- 고객의 호환 가능한 모든 클러스터에 할당할 수 있습니다.
- 고객이 사용자 지정 캐시 파라미터 그룹을 생성하기 위해 수정할 수 있습니다.

모든 파라미터 값을 수정할 수 있는 것은 아닙니다. 자세한 내용은 [Memcached 특정 파라미터 \(p. 127\)](#) 단원을 참조하십시오.

## 파라미터 그룹 생성

기본값에서 변경하려는 파라미터 값이 하나 이상이면 새 파라미터 그룹을 생성해야 합니다. ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 파라미터 그룹을 생성할 수 있습니다.

## 파라미터 그룹 생성 (콘솔)

다음 절차는 ElastiCache 콘솔을 사용하여 파라미터 그룹을 생성하는 방법을 보여줍니다.

ElastiCache 콘솔을 사용하여 파라미터 그룹을 생성하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 사용 가능한 모든 파라미터 목록을 표시하려면 왼쪽 탐색 창에서 [Parameter Groups]를 선택합니다.
3. 파라미터 그룹을 생성하려면 [Create Parameter Group]을 선택합니다.

[Create Parameter Group] 화면이 나타납니다.

4. [Family] 목록에서 파라미터 그룹의 템플릿이 될 파라미터 그룹 패밀리를 선택합니다.

memcached1.4 같은 파라미터 그룹 패밀리는 파라미터 그룹의 실제 파라미터와 초기 값을 정의합니다. 파라미터 그룹 패밀리는 클러스터의 엔진 및 버전과 일치해야 합니다.

5. [Name] 상자에 파라미터 그룹의 고유 이름을 입력합니다.

클러스터를 생성하거나 클러스터의 파라미터 그룹을 수정할 때 그 이름으로 파라미터 그룹을 선택합니다. 그러므로 이름은 파라미터 그룹의 패밀리를 식별할 수 있고 정보를 알 수 있는 것이 좋습니다.

파라미터 그룹 명명 제약 조건은 다음과 같습니다.

- ASCII 문자로 시작해야 합니다.
- ASCII 문자, 숫자 및 하이픈만 포함할 수 있습니다.
- 1-255자여야 합니다.
- 하이픈 2개가 연속될 수 없습니다.
- 끝에 하이픈이 올 수 없습니다.

6. [Description] 상자에 파라미터 그룹에 대한 설명을 입력합니다.
7. 파라미터 그룹을 생성하려면 [Create]를 선택합니다.

파라미터 그룹을 생성하지 않고 프로세스를 종료하려면 [Cancel]을 선택합니다.

8. 파라미터 그룹을 생성하면 패밀리의 기본값이 부여됩니다. 기본값을 변경하려면 파라미터 그룹을 수정해야 합니다. 자세한 내용은 [파라미터 그룹 수정 \(p. 124\)](#) 단원을 참조하십시오.

## 파라미터 그룹 생성 (AWS CLI)

AWS CLI를 사용하여 파라미터 그룹을 생성하려면 이 파라미터와 함께 `create-cache-parameter-group` 명령을 사용합니다.

- `--cache-parameter-group-name` — 파라미터 그룹의 이름입니다.

파라미터 그룹 명명 제약 조건은 다음과 같습니다.

- ASCII 문자로 시작해야 합니다.
- ASCII 문자, 숫자 및 하이픈만 포함할 수 있습니다.
- 1-255자여야 합니다.
- 하이픈 2개가 연속될 수 없습니다.
- 끝에 하이픈이 올 수 없습니다.

- `--cache-parameter-group-family` — 파라미터 그룹의 엔진 및 버전 패밀리입니다.

- `--description` — 사용자가 정의한 파라미터 그룹에 대한 설명입니다.

## Example

다음 예제에서는 memcached1.4를 템플릿으로 사용하여 myMem14라는 파라미터 그룹을 생성합니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache create-cache-parameter-group \
 --cache-parameter-group-name myMem14 \
 --cache-parameter-group-family memcached1.4 \
 --description "My first parameter group"
```

Windows의 경우:

```
aws elasticache create-cache-parameter-group ^
 --cache-parameter-group-name myMem14 ^
 --cache-parameter-group-family memcached1.4 ^
 --description "My first parameter group"
```

이 명령의 출력은 다음과 유사해야 합니다.

```
{
 "CacheParameterGroup": {
 "CacheParameterGroupName": "myMem14",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
}
```

파라미터 그룹을 생성하면 패밀리의 기본값이 부여됩니다. 기본값을 변경하려면 파라미터 그룹을 수정해야 합니다. 자세한 내용은 [파라미터 그룹 수정 \(p. 124\)](#) 단원을 참조하십시오.

자세한 내용은 [create-cache-parameter-group](#) 단원을 참조하십시오.

## 파라미터 그룹 생성 (ElastiCache API)

ElastiCache API를 사용하여 파라미터 그룹을 생성하려면 이 파라미터와 함께 CreateCacheParameterGroup 작업을 사용합니다.

- **ParameterGroupName** — 파라미터 그룹의 이름입니다.  
파라미터 그룹 명명 제약 조건은 다음과 같습니다.
  - ASCII 문자로 시작해야 합니다.
  - ASCII 문자, 숫자 및 하이픈만 포함할 수 있습니다.
  - 1-255자여야 합니다.
  - 하이픈 2개가 연속될 수 없습니다.
  - 끝에 하이픈이 올 수 없습니다.
- **CacheParameterGroupFamily** — 파라미터 그룹의 엔진 및 버전 패밀리입니다. 예: memcached1.4.
- **Description** — 사용자가 정의한 파라미터 그룹에 대한 설명입니다.

## Example

다음 예제에서는 memcached1.4를 템플릿으로 사용하여 myMem14라는 파라미터 그룹을 생성합니다.

```
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>
```

이 작업의 응답은 다음과 유사해야 합니다.

```
<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>
```

파라미터 그룹을 생성하면 패밀리의 기본값이 부여됩니다. 기본값을 변경하려면 파라미터 그룹을 수정해야 합니다. 자세한 내용은 [파라미터 그룹 수정 \(p. 124\)](#) 단원을 참조하십시오.

자세한 내용은 [CreateCacheParameterGroup](#) 단원을 참조하십시오.

## 이름별로 파라미터 그룹 목록 조회

ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 파라미터 그룹을 나열할 수 있습니다.

### 이름으로 파라미터 그룹 나열 (콘솔)

다음 절차에서는 ElastiCache 콘솔을 사용하여 파라미터 그룹 목록을 보는 방법을 설명합니다.

ElastiCache 콘솔을 사용하여 파라미터 그룹을 나열하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 사용 가능한 모든 파라미터 목록을 표시하려면 왼쪽 탐색 창에서 [Parameter Groups]를 선택합니다.

### 이름으로 파라미터 그룹 나열 (AWS CLI)

AWS CLI를 사용하여 파라미터 그룹의 목록을 생성하려면 `describe-cache-parameter-groups` 명령을 사용합니다. 파라미터 그룹의 이름을 입력하면 그 파라미터 그룹만 나열됩니다. 파라미터 그룹의 이름을 입력하지 않으면 최대 `--max-records`개의 파라미터 그룹이 나열됩니다. 두 경우 모두 파라미터 그룹의 이름, 패밀리 및 설명이 나열됩니다.

#### Example

다음 샘플 코드에는 `myMem14` 파라미터 그룹이 나열되어 있습니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache describe-cache-parameter-groups \
 --cache-parameter-group-name myMem14
```

Windows의 경우:

```
aws elasticache describe-cache-parameter-groups ^
--cache-parameter-group-name myMem14
```

이 명령의 출력은 파라미터 그룹의 이름, 패밀리 및 설명을 나열하는 것과 같습니다.

```
{
 "CacheParameterGroups": [
 {
 "CacheParameterGroupName": "myMem14",
 "CacheParameterGroupFamily": "memcached1.4",
 "Description": "My first parameter group"
 }
]
}
```

### Example

다음 샘플 코드는 최대 10개의 파라미터 그룹을 나열합니다.

```
aws elasticache describe-cache-parameter-groups --max-records 20
```

이 명령의 JSON 출력은 각 파라미터 그룹의 이름, 패밀리, 설명 및 redis5.6의 경우 파라미터 그룹이 글로벌 데이터 스토어(IsGlobal)의 일부인지 여부를 나열하는 것과 같습니다.

```
{
 "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": true
 }
]
}
```

```
 "isGlobal": "yes"
 },
]
}
```

자세한 내용은 [describe-cache-parameter-groups](#) 단원을 참조하십시오.

## 이름으로 파라미터 그룹 나열 (ElastiCache API)

ElastiCache API를 사용하여 파라미터 그룹의 목록을 생성하려면 DescribeCacheParameterGroups 작업을 사용합니다. 파라미터 그룹의 이름을 입력하면 그 파라미터 그룹만 나열됩니다. 파라미터 그룹의 이름을 입력하지 않으면 최대 MaxRecords개의 파라미터 그룹이 나열됩니다. 두 경우 모두 파라미터 그룹의 이름, 패밀리 및 설명이 나열됩니다.

### Example

다음 샘플 코드에는 myMem14 파라미터 그룹이 나열되어 있습니다.

```
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>
```

이 작업의 응답은 각 파라미터 그룹의 이름, 패밀리 및 설명을 나열하는 것과 같습니다.

```
<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

다음 샘플 코드는 최대 10개의 파라미터 그룹을 나열합니다.

```
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>
```

이 작업의 응답은 각 파라미터 그룹의 이름, 패밀리, 설명 및 redis5.6의 경우 파라미터 그룹이 글로벌 데이터 스토어(IsGlobal)의 일부인지 여부를 나열하는 것과 같습니다.

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
이름별로 파라미터 그룹 목록 조회

```
<DescribeCacheParameterGroupsResponse xmlns="http://elasticache.amazonaws.com/doc/2013-06-15/">
 <DescribeCacheParameterGroupsResult>
 <CacheParameterGroups>
 <CacheParameterGroup>
 <CacheParameterGroupName>myRedis28</CacheParameterGroupName>
 <CacheParameterGroupFamily>redis2.8</CacheParameterGroupFamily>
 <Description>My custom Redis 2.8 parameter group</Description>
 </CacheParameterGroup>
 <CacheParameterGroup>
 <CacheParameterGroupName>myMem14</CacheParameterGroupName>
 <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
 <Description>My custom Memcached 1.4 parameter group</Description>
 </CacheParameterGroup>
 <CacheParameterGroup>
 <CacheParameterGroupName>myRedis56</CacheParameterGroupName>
 <CacheParameterGroupFamily>redis5.0</CacheParameterGroupFamily>
 <Description>My custom redis 5.6 parameter group</Description>
 <isGlobal>yes</isGlobal>
 </CacheParameterGroup>
 </CacheParameterGroups>
 </DescribeCacheParameterGroupsResult>
 <ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

### Example

다음 샘플 코드는 최대 10개의 파라미터 그룹을 나열합니다.

```
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>
```

이 작업의 응답은 각 파라미터 그룹의 이름, 패밀리, 설명 및 redis5.6의 경우 파라미터 그룹이 글로벌 데이터 스토어(IsGlobal)의 일부인지 여부를 나열하는 것과 같습니다.

```
<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>
 </CacheParameterGroup>
 </CacheParameterGroups>
 </DescribeCacheParameterGroupsResult>
 <ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

```
<isGlobal>yes</isGlobal>
</CacheParameterGroup>
</DescribeCacheParameterGroupsResult>
<ResponseMetadata>
 <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

자세한 내용은 [DescribeCacheParameterGroups](#) 단원을 참조하십시오.

## 파라미터 그룹의 값 목록 조회

ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 파라미터 및 파라미터 그룹의 값을 나열할 수 있습니다.

### 파라미터 그룹 값 나열 (콘솔)

다음 절차에서는 ElastiCache 콘솔을 사용하여 파라미터 및 파라미터 그룹의 값을 나열하는 방법을 보여줍니다.

ElastiCache 콘솔을 사용하여 파라미터 그룹의 파라미터 및 그 값을 나열하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 사용 가능한 모든 파라미터 목록을 표시하려면 왼쪽 탐색 창에서 [Parameter Groups]를 선택합니다.
3. 파라미터 그룹의 이름 왼쪽에 있는 확인란을 선택하여 파라미터 및 값을 나열할 파라미터 그룹을 선택합니다.

파라미터 및 그 값이 화면 하단에 나열됩니다. 파라미터의 수가 많으면 위아래로 스크롤하여 원하는 파라미터를 찾습니다.

### 파라미터 그룹 값 나열 (AWS CLI)

AWS CLI를 사용하여 파라미터 그룹의 파라미터 및 그 값을 나열하려면 `describe-cache-parameters` 명령을 사용합니다.

#### Example

다음 샘플 코드에는 myMem14 파라미터 그룹의 모든 파라미터 및 그 값이 나열되어 있습니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache describe-cache-parameters \
 --cache-parameter-group-name myMem14
```

Windows의 경우:

```
aws elasticache describe-cache-parameters ^
 --cache-parameter-group-name myMem14
```

자세한 내용은 [describe-cache-parameters](#) 단원을 참조하십시오.

### 파라미터 그룹 값 나열 (ElastiCache API)

ElastiCache API를 사용하여 파라미터 그룹의 파라미터 및 그 값을 나열하려면 `DescribeCacheParameters` 작업을 사용합니다.

## Example

다음 샘플 코드에는 myMem14 파라미터 그룹의 모든 파라미터가 나열되어 있습니다.

```
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>
```

이 작업의 응답은 다음과 유사합니다. 응답이 잘렸습니다.

```
<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>
 </CacheNodeTypeSpecificParameter>
 </CacheClusterClassSpecificParameters>
 </DescribeCacheParametersResult>
 <ResponseMetadata>
 <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
 </ResponseMetadata>
 </DescribeCacheParametersResponse>
```

자세한 내용은 [DescribeCacheParameters](#) 단원을 참조하십시오.

## 파라미터 그룹 수정

### Important

어떤 기본 파라미터 그룹도 수정할 수 없습니다.

파라미터 그룹의 일부 파라미터 값을 수정할 수 있습니다. 이 파라미터 값은 파라미터 그룹과 연결된 클러스터에 적용됩니다. 변경한 파라미터 값이 파라미터 그룹에 적용되는 시점에 관한 자세한 내용은 [Memcached 특정 파라미터 \(p. 127\)](#) 단원을 참조하십시오.

## 파라미터 그룹 수정 (콘솔)

다음 절차는 ElastiCache 콘솔을 사용하여 `binding_protocol` 파라미터 값을 변경하는 방법을 보여줍니다. 동일한 절차를 통해 모든 파라미터 값을 변경할 수 있습니다.

ElastiCache 콘솔을 사용하여 파라미터 값을 변경하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 사용 가능한 모든 파라미터 목록을 표시하려면 왼쪽 탐색 창에서 [Parameter Groups]를 선택합니다.
3. 파라미터 그룹의 이름 왼쪽에 있는 확인란을 선택하여 수정할 파라미터 그룹을 선택합니다.  
  
파라미터 그룹의 파라미터가 화면 하단에 나열됩니다. 모든 파라미터를 보려면 목록 페이지를 탐색해야 할 수도 있습니다.
4. 파라미터를 하나 이상 수정하려면 [Edit Parameters]를 선택합니다.
5. [Edit Parameter Group:] 화면에서 왼쪽/오른쪽 화살표로 스크롤하여 `binding_protocol` 파라미터를 찾은 다음 [Value] 열에 [ascii]를 입력합니다.
6. [Save Changes]를 선택합니다.
7. 변경한 파라미터의 이름을 찾으려면 [Memcached 특정 파라미터 \(p. 127\)](#) 단원을 참조하십시오. 다시 시작한 후 파라미터에 변경 사항이 발생하면 이 파라미터 그룹을 사용하는 모든 클러스터를 재부팅합니다. 자세한 내용은 [클러스터 재부팅 \(p. 77\)](#) 단원을 참조하십시오.

## 파라미터 그룹 수정 (AWS CLI)

AWS CLI를 사용하여 파라미터 값을 변경하려면 `modify-cache-parameter-group` 명령을 사용합니다.

### Example

변경하려는 파라미터의 이름과 허용되는 값을 찾으려면 [Memcached 특정 파라미터 \(p. 127\)](#) 단원을 참조하십시오.

다음 샘플 코드에서는 파라미터 그룹 `myMem14`에 대해 두 파라미터, 즉 `chunk_size` 및 `chunk_size_growth_fact`의 값을 설정합니다.

Linux, macOS, or Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

이 명령의 출력은 다음과 같습니다.

```
{
 "CacheParameterGroupName": "myMem14"
}
```

자세한 내용은 [modify-cache-parameter-group](#) 단원을 참조하십시오.

다시 시작한 후 파라미터에 변경 사항이 발생하면 이 파라미터 그룹을 사용하는 모든 클러스터를 재부팅합니다. 자세한 내용은 [클러스터 재부팅 \(p. 77\)](#) 단원을 참조하십시오.

## 파라미터 그룹 수정 (ElastiCache API)

ElastiCache API를 사용하여 파라미터 그룹의 파라미터 값을 변경하려면 `ModifyCacheParameterGroup` 작업을 사용합니다.

### Example

변경하려는 파라미터의 이름과 허용되는 값을 찾으려면 [Memcached 특정 파라미터 \(p. 127\)](#) 단원을 참조하십시오.

다음 샘플 코드에서는 파라미터 그룹 `myMem14`에 대해 두 파라미터, 즉 `chunk_size` 및 `chunk_size_growth_fact`의 값을 설정합니다.

```
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>
```

자세한 내용은 [ModifyCacheParameterGroup](#) 단원을 참조하십시오.

파라미터를 업데이트하고 저장한 후에 파라미터에 변경 사항이 발생하면 다시 시작한 후에 이 파라미터 그룹을 사용하는 각 클러스터를 재부팅합니다. 자세한 내용은 [클러스터 재부팅 \(p. 77\)](#) 단원을 참조하십시오.

## 파라미터 그룹 삭제

ElastiCache 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 사용자 지정 파라미터 그룹을 삭제할 수 있습니다.

파라미터 그룹이 클러스터와 연결된 경우 삭제할 수 없습니다. 또한 기본 파라미터 그룹도 삭제할 수 없습니다.

### 파라미터 그룹 삭제 (콘솔)

다음 절차는 ElastiCache 콘솔을 사용하여 파라미터 그룹을 삭제하는 방법을 보여줍니다.

ElastiCache 콘솔을 사용하여 파라미터 그룹을 삭제하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 사용 가능한 모든 파라미터 목록을 표시하려면 왼쪽 탐색 창에서 [Parameter Groups]를 선택합니다.
3. 파라미터 그룹의 이름 왼쪽에 있는 확인란을 선택하여 삭제할 파라미터 그룹을 선택합니다.

[Delete] 버튼이 활성화됩니다.

4. 삭제를 선택합니다.

[Delete Parameter Groups] 확인 화면이 나타납니다.

5. 파라미터 그룹을 삭제하려면 [Delete Parameter Groups] 확인 화면에서 [Delete]를 추가합니다.  
파라미터 그룹을 유지하려면 [Cancel]을 선택합니다.

## 파라미터 그룹 삭제 (AWS CLI)

`delete-cache-parameter-group`를 사용하여 파라미터 그룹을 삭제하려면 AWS CLI 명령을 사용합니다. 삭제할 파라미터 그룹의 경우 `--cache-parameter-group-name`으로 지정된 파라미터 그룹에는 클러스터를 연결할 수 없으며 기본 파라미터 그룹이 될 수도 없습니다.

다음 샘플 코드에서는 `myMem14` 파라미터 그룹을 삭제합니다.

### Example

Linux, macOS, or Unix의 경우:

```
aws elasticache delete-cache-parameter-group \
 --cache-parameter-group-name myMem14
```

Windows의 경우:

```
aws elasticache delete-cache-parameter-group ^
 --cache-parameter-group-name myMem14
```

자세한 내용은 `delete-cache-parameter-group` 단원을 참조하십시오.

## 파라미터 그룹 삭제 (ElastiCache API)

ElastiCache API를 사용하여 파라미터 그룹을 삭제하려면 `DeleteCacheParameterGroup` 작업을 사용합니다. 삭제할 파라미터 그룹의 경우 `CacheParameterGroupName`으로 지정된 파라미터 그룹에는 클러스터를 연결할 수 없으며 기본 파라미터 그룹이 될 수도 없습니다.

### Example

다음 샘플 코드에서는 `myMem14` 파라미터 그룹을 삭제합니다.

```
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>
```

자세한 내용은 `DeleteCacheParameterGroup` 단원을 참조하십시오.

## Memcached 특정 파라미터

Memcached 클러스터에 파라미터 그룹을 지정하지 않으면 엔진 버전에 적절한 기본 파라미터 그룹이 사용 됩니다. 기본 파라미터 그룹에서는 어떤 파라미터 값도 변경할 수 없습니다. 그러나 사용자 지정 파라미터 그룹을 생성하여 언제든지 클러스터에 할당할 수 있습니다. 자세한 내용은 [파라미터 그룹 생성 \(p. 116\)](#) 단원을 참조하십시오.

### 주제

- [Memcached 1.5.10 파라미터 변경 \(p. 128\)](#)
- [Memcached 1.4.34 추가 파라미터 \(p. 129\)](#)

- [Memcached 1.4.33 추가 파라미터 \(p. 129\)](#)
- [Memcached 1.4.24 추가 파라미터 \(p. 130\)](#)
- [Memcached 1.4.14 추가 파라미터 \(p. 131\)](#)
- [Memcached 1.4.5 지원 파라미터 \(p. 135\)](#)
- [Memcached 연결 오버헤드 \(p. 136\)](#)
- [Memcached 노드 유형별 파라미터 \(p. 137\)](#)

## Memcached 1.5.10 파라미터 변경

Memcached 1.5.10은 다음과 같은 추가 파라미터를 지원합니다.

파라미터 그룹 Family: memcached1.5

이름	세부 정보	설명
no_modern	기본값: 0 유형: 부울 수정 가능 여부: 예 허용된 값: 0,1 변경 사항 적용: 시작 시	slab_reassign, slab_automove, lru_crawler, lru_maintainer, maxconns_fast 명령 비활성화를 위한 별칭. No modern은 hash_algorithm을 jenkins로 설정하고 ASCII VALUE의 인라이닝을 허용합니다. memcached 1.5 이상에 적용됩니다. 현재 기본값인 modern으로 되돌리려면 다시 시작해야 합니다.
inline_ascii_resp	기본값: 0 유형: 부울 수정 가능 여부: 예 허용된 값: 0,1 변경 사항 적용: 시작 시	최대 24바이트를 사용하여 항목 내 VALUE 응답의 수치를 저장합니다. ASCII get, faster 세트의 속도가 약간 느려집니다.

Memcached 1.5.10의 경우 다음과 같은 파라미터가 제거됩니다.

이름	세부 정보	설명
expirezero_does_not_evict	기본값: 0 유형: 부울 수정 가능 여부: 예 허용된 값: 0,1 변경 사항 적용: 시작 시	이 버전에서는 이제 지원하지 않습니다.
modern	기본값: 1 유형: 부울 수정 가능 여부: 예(no-modern으로 설정하는 경우 재시작해야 함)	이 버전에서는 이제 지원하지 않습니다. 이 버전부터는 시작할 때마다 항상 또는 재시작 시 기본적으로 modern이 활성화됩니다.

이름	세부 정보	설명
	허용된 값: 0,1 변경 사항 적용: 시작 시	

## Memcached 1.4.34 추가 파라미터

Memcached 1.4.34는 추가 파라미터를 지원하지 않습니다.

파라미터 그룹 패밀리: memcached1.4

## Memcached 1.4.33 추가 파라미터

For Memcached 1.4.33은 다음과 같은 추가 파라미터를 지원합니다.

파라미터 그룹 패밀리: memcached1.4

이름	세부 정보	설명
modern	기본값: 활성화 유형: 부울 수정 가능 여부: 예 변경 사항 적용: 시작 시	여러 기능의 별칭입니다. modern을 활성화하는 것은 murmur3 해시 알고리즘을 사용하고 다음 명령을 사용하는 것과 같습니다. slab_reassign, slab_automove, lru_crawler, lru_maintainer, maxconns_fast 및 hash_algorithm=murmur3
watch	기본값: 활성화 유형: 부울 수정 가능 여부: 예 변경 사항 적용: 즉시 사용자가 watcher_logbuf_size 및 worker_logbuf_size 한도에 도달하면 로그가 삭제될 수 있습니다.	로그 가져오기, 제거 또는 변형. 예를 들어 사용자가 watch를 켜면 get, set, delete 또는 update 발생 시 로그를 볼 수 있습니다.
idle_timeout	기본값: 0(비활성화) 유형: 정수 수정 가능 여부: 예 변경 사항 적용: 시작 시	종료하라는 메시지가 표시되기 전에 클라이언트가 유휴 상태로 있을 수 있는 최소 시간(초)입니다. 값의 범위는 0~86400입니다.
track_sizes	기본값: 비활성화 유형: 부울 수정 가능 여부: 예 변경 사항 적용: 시작 시	슬래브 그룹이 소비한 크기를 표시합니다.  track_sizes를 활성화하면 stats sizes_enable을 실행할 필요 없이 stats sizes를 실행할 수 있습니다.

이름	세부 정보	설명
watcher_logbuf_size	기본값: 256(KB) 유형: 정수 수정 가능 여부: 예 변경 사항 적용: 시작 시	watch 명령은 Memcached에 대한 스트림 로깅을 켭니다. 그러나 로깅 버퍼가 가득 찰 정도로 제거, 변형 또는 가져오기 비율이 높은 경우 watch에서 로그를 삭제할 수 있습니다. 이러한 상황에서 사용자는 로그 손실을 줄이기 위해 버퍼 크기를 늘릴 수 있습니다.
worker_logbuf_size	기본값: 64(KB) 유형: 정수 수정 가능 여부: 예 변경 사항 적용: 시작 시	watch 명령은 Memcached에 대한 스트림 로깅을 켭니다. 그러나 버퍼가 가득 찰 정도로 제거, 변형 또는 가져오기 비율이 높으면 watch는 로그를 삭제할 수 있습니다. 이러한 상황에서 사용자는 로그 손실을 줄이기 위해 버퍼 크기를 늘릴 수 있습니다.
slab_chunk_max	기본값: 524288(바이트) 유형: 정수 수정 가능 여부: 예 변경 사항 적용: 시작 시	슬래브의 최대 크기를 지정합니다. 슬래브 크기를 작게 설정하면 메모리를 더 효율적으로 사용합니다. slab_chunk_max보다 큰 항목은 여러 슬래브로 분할됩니다.
lru_crawler metadump [all 1 2 3]	기본값: 비활성화 유형: 부울 수정 가능 여부: 예 변경 사항 적용: 즉시	lru_crawler를 활성화하면 이 명령이 모든 키를 덤프합니다.  all 1 2 3 - 모든 슬래브 또는 특정 슬래브 수 지정

## Memcached 1.4.24 추가 파라미터

Memcached 1.4.24는 다음과 같은 추가 파라미터를 지원합니다.

파라미터 그룹 패밀리: memcached1.4

이름	세부 정보	설명
disable_flush_all	기본값: 0(비활성화) 유형: 부울 수정 가능 여부: 예 변경 사항 적용: 시작 시	flush_all을 비활성화하려면 파라미터 (-F)를 추가합니다. 프로덕션 인스턴스에서 전체 플러시를 실행할 수 없는 경우에 유용합니다.  값은 0, 1(값이 0일 때 사용자가 flush_all을 수행할 수 있음)입니다.
hash_algorithm	기본값: jenkins 유형: 문자열 수정 가능 여부: 예 변경 사항 적용: 시작 시	사용할 해시 알고리즘입니다. 허용되는 값은 murmur3 및 jenkins입니다.

이름	세부 정보	설명
lru_crawler	<p>기본값: 0(비활성화)</p> <p>유형: 부울</p> <p>수정 가능 여부: 예</p> <p>변경 사항 적용: 재시작 후</p> <p><b>Note</b></p> <p>런타임 시 명령줄에서 lru_crawler를 일시적으로 활성화할 수 있습니다. 자세한 내용은 Description(설명) 열을 참조하십시오.</p>	<p>만료된 항목의 슬래브 클래스를 삭제합니다. 백그라운드에서 실행되는 영향이 적은 프로세스입니다. 현재는 수동 명령을 사용하여 크롤링을 시작해야 합니다.</p> <p>일시적으로 활성화하려면 명령줄에서 lru_crawler enable을 실행합니다.</p> <p>lru_crawler 1,3,5는 슬래브 클래스 1, 3 및 5를 크롤링하여 freelist에 추가할 만료 항목을 찾습니다.</p> <p>값: 0,1</p> <p><b>Note</b></p> <p>명령줄에서 lru_crawler를 활성화하면 명령줄에서 비활성화하거나 다음에 재부팅될 때까지 크롤러가 활성화됩니다. 영구적으로 활성화하려면 파라미터 값을 수정해야 합니다. 자세한 내용은 <a href="#">파라미터 그룹 수정 (p. 124)</a> 단원을 참조하십시오.</p>
lru_maintainer	<p>기본값: 0(비활성화)</p> <p>유형: 부울</p> <p>수정 가능 여부: 예</p> <p>변경 사항 적용: 시작 시</p>	<p>용량에 도달할 때 LRU 간에 항목을 셔플링한 백그라운드 스레드입니다. 값: 0, 1</p>
expirezero_does_not_evict	<p>기본값: 0(비활성화)</p> <p>유형: 부울</p> <p>수정 가능 여부: 예</p> <p>변경 사항 적용: 시작 시</p>	<p>lru_maintainer와 함께 사용하면 만료 시간이 0인 항목을 제거할 수 없게 합니다.</p> <p><b>Warning</b></p> <p>기타 제거할 수 있는 항목에 사용 가능한 메모리를 밀어낼 수 있습니다.</p> <p>lru_maintainer를 무시하도록 설정할 수 있습니다.</p>

## Memcached 1.4.14 추가 파라미터

Memcached 1.4.14는 다음과 같은 추가 파라미터를 지원합니다.

파라미터 그룹 패밀리: memcached1.4

Memcached 1.4.14에 추가된 파라미터

이름	설명
config_max	<p>ElastiCache 구성 항목의 최대 수입니다.</p> <p>기본값: 16</p> <p>유형: 정수</p> <p>수정 가능 여부: 아니요</p>
config_size_max	<p>구성 항목의 최대 크기(바이트)입니다.</p> <p>기본값: 65536</p> <p>유형: 정수</p> <p>수정 가능 여부: 아니요</p>
hashpower_init	<p>ElastiCache 해시 테이블의 처음 크기이며 2의 거듭 제곱으로 표시됩니다. 기본값은 <math>16(2^{16})</math> 또는 65536 키입니다.</p> <p>기본값: 16</p> <p>유형: 정수</p> <p>수정</p>

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
Memcached 특정 파라미터

이름	설명
	가 능 여 부: 아 니 요
maxconns_fast	<p>최대 연결 한도에 도달했을 때 새 연결 요청을 처리하는 방식을 변경합니다. 이 파라미터를 0으로 설정하면 새 연결이 백로그 대기열에 추가되고 다른 연결이 끝날 때까지 대기합니다. 파라미터를 1로 설정하면 ElastiCache가 클라이언트 요청을 전송하고 즉시 연결을 끊습니다.</p> <p>유 형: 부 울</p> <p>수 정 가 능 여 부: 예</p> <p>변 경 사 항 용: 시 작 후</p> <p>적 재</p>

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
Memcached 특정 파라미터

이름	설명
slab_automove	<p>슬래브 오토무브 알고리즘을 조정합니다. 이 파라미터를 0(영)으로 설정하면 오토무브 알고리즘이 비활성화됩니다. 1로 설정하면 ElastiCache가 자동으로 슬래브를 이동하는 데 느리고 보수적인 접근 방식을 취합니다. 2로 설정하면 제거할 때마다 ElastiCache가 적극적으로 슬래브를 이동합니다. (이 모드는 테스트 목적을 제외하고는 권장되지 않음)</p> <p>유형: 정수</p> <p>수정 가능 여부: 예</p> <p>변경 사항 적용: 재시작 후</p>
slab_reassign	<p>슬래브 재할당을 활성화하거나 비활성화합니다. 이 파라미터를 1로 설정하면 "슬래브 재할당" 명령을 사용하여 메모리를 수동으로 재할당할 수 있습니다.</p> <p>값: 0 (false)</p> <p>유형: 부울</p> <p>수정 가능 여부: 예</p> <p>변경 사항 적용: 재시작 후</p>

## Memcached 1.4.5 지원 파라미터

파라미터 그룹 패밀리: memcached1.4

Memcached 1.4.5는 다음과 같은 파라미터를 지원합니다.

Memcached 1.4.5에 추가된 파라미터

이름	세부 정보	설명
backlog_queue_size	기본값: 1024 유형: 정수 수정 가능 여부: 아니요	백 로그 대기열 제한입니다.
binding_protocol	기본값: 자동 유형: 문자열 수정 가능 여부: 예 변경 사항 적용: 재시작 후	바인딩 프로토콜입니다. 허용 가능한 값은 ascii 및 auto입니다.  binding_protocol의 값을 수정하는 방법에 대한 지침은 <a href="#">파라미터 그룹 수정 (p. 124)</a> 을 참조하십시오.
cas_disabled	기본값: 0 (false) 유형: 부울 수정 가능 여부: 예 변경 사항 적용: 재시작 후	1(true)이면 확인 및 설정(CAS) 작업이 비활성화되고 저장된 항목이 CAS가 활성화된 경우보다 8바이트 적게 소비합니다.
chunk_size	기본값: 48 유형: 정수 수정 가능 여부: 예 변경 사항 적용: 재시작 후	가장 작은 항목의 키, 값 및 플래그에 할당할 공간의 최소 크기(바이트)입니다.
chunk_size_growth_factor	기본값: 25 유형: float 수정 가능 여부: 예 변경 사항 적용: 재시작 후	연속된 Memcached 청크 크기를 제어하는 성장 인자입니다. 각 청크는 이전 청크보다 chunk_size_growth_factor배 더 큼니다.
error_on_memory_exhaustion	기본값: 0 (false) 유형: 부울 수정 가능 여부: 예 변경 사항 적용: 재시작 후	1(true)이면 항목을 저장할 메모리가 없으면 Memcached가 항목을 제거하는 대신 오류를 반환합니다.
large_memory_pages	기본값: 0 (false) 유형: 부울 수정 가능 여부: 아니요	1(true)이면 ElastiCache가 더 큰 메모리 페이지를 사용하고자 합니다.

이름	세부 정보	설명
lock_down_page	기본값: 0 (false) 유형: 부울 수정 가능 여부: 아니요	1(true)이면 ElastiCache가 메이징된 모든 메모리를 잠급니다.
max_item_size	기본값: 1048576 유형: 정수 수정 가능 여부: 예 변경 사항 적용: 재시작 후	클러스터에 저장할 수 있는 가장 큰 항목의 크기(바이트)입니다.
max_simultaneous_connections	기본값: 65536 유형: 정수 수정 가능 여부: 아니요	최대 동시 연결 수입니다.
maximize_core_file_size	기본값: 0 (false) 유형: 부울 수정 가능: 변경 사항 적용: 없음	1(true)이면 ElastiCache가 핵심 파일 제한을 최대화합니다.
memcached_connections_overhead	기본값: 100 유형: 정수 수정 가능 여부: 예 변경 사항 적용: 재시작 후	Memcached 연결 및 기타 오버헤드에 예약된 메모리 양입니다. 이 파라미터에 대한 자세한 내용은 <a href="#">Memcached 연결 오버헤드 (p. 136)</a> 를 참조하십시오.
requests_per_event	기본값: 20 유형: 정수 수정 가능 여부: 아니요	지정된 연결의 이벤트 당 최대 요청 수입니다. 이 제한은 리소스 결핍을 막기 위해 필요합니다.

## Memcached 연결 오버헤드

각 노드에서 항목을 저장하는 데 사용할 수 있는 메모리는 max\_cache\_memory 파라미터에 저장된 해당 노드의 총 사용 가능한 메모리에서 memcached\_connections\_overhead 파라미터에 저장된 연결에 사용하는 메모리와 기타 오버헤드를 뺀 값입니다. 예를 들어, cache.m1.small 유형의 노드에는 1300MB의 max\_cache\_memory가 있습니다. 기본 memcached\_connections\_overhead 값이 100MB이면 Memcached 프로세스는 항목을 저장하는 데 1200MB를 사용할 수 있습니다.

memcached\_connections\_overhead 파라미터의 기본값은 대부분의 사용 사례를 충족시키지만 연결 오버헤드에 필요한 할당량은 요청 빈도, 페이로드 크기 및 연결 수를 비롯한 여러 요소에 따라 달라질 수 있습니다.

애플리케이션에 맞게 memcached\_connections\_overhead 값을 변경할 수 있습니다. 예를 들어, memcached\_connections\_overhead 파라미터 값을 높이면 항목을 저장하는 데 사용할 수 있는 메모리 양이 줄어들어 연결 오버헤드에 더 큰 버퍼가 제공됩니다. memcached\_connections\_overhead 파라미터 값을 줄이면 항목을 저장하는 데 더 많은 메모리를 사용할 수 있지만 스왑 사용량 및 성능 저하 위험이 높

아질 수 있습니다. 스왑 사용량이 늘고 성능 저하가 발생하면 `memcached_connections_overhead` 파라미터 값을 늘립니다.

#### Important

`cache.t1.micro` 노드 유형의 경우 `memcached_connections_overhead` 값은 다음과 같이 결정됩니다.

- 클러스터가 기본 파라미터 그룹을 사용하면 ElastiCache는 `memcached_connections_overhead` 값을 13MB로 설정합니다.
- 클러스터가 사용자가 직접 생성한 파라미터 그룹을 사용하면 `memcached_connections_overhead` 값을 원하는 대로 설정할 수 있습니다.

## Memcached 노드 유형별 파라미터

대부분의 파라미터는 단일 값을 갖지만 일부 파라미터는 사용하는 노드 유형에 따라 다양한 값을 갖습니다. 다음 표에는 각 노드 유형에 대한 `max_cache_memory` 및 `num_threads` 파라미터의 기본값이 나와 있습니다. 이 파라미터의 값은 수정할 수 없습니다.

노드 유형	max_cache_memory(MB)	num_threads
cache.t1.micro	213	1
cache.t2.micro	555	1
cache.t2.small	1588	1
cache.t2.medium	3301	2
cache.t3.micro	512	2
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.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	14618	4

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
Memcached 특정 파라미터

노드 유형	max_cache_memory(MB)	num_threads
cache.m4.2xlarge	30412	8
cache.m4.4xlarge	62234	16
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.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
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

Note

모든 T2 인스턴스는 Amazon Virtual Private Cloud(Amazon VPC)에서 생성됩니다.

# 사용 내역, 이벤트 및 비용 모니터링

엔터프라이즈 캐싱 솔루션을 관리하려면 클러스터가 어떻게 수행하고 클러스터에서 사용하는 리소스를 알아야 합니다. 또한 생성되는 이벤트와 배포 비용을 파악하는 것도 중요합니다.

Amazon CloudWatch는 캐시 성능을 모니터링하기 위한 측정치를 제공합니다. 또한, 비용 할당 태그는 비용을 모니터링하고 관리하는 데 도움이 됩니다.

## 주제

- [측정치를 통한 모니터링 사용 \(p. 139\)](#)
- [ElastiCache 이벤트 모니터링 \(p. 145\)](#)
- [비용 할당 태그를 사용한 비용 모니터링 \(p. 153\)](#)
- [예약 노드를 통해 비용 관리 \(p. 159\)](#)

## 측정치를 통한 모니터링 사용

클러스터를 모니터링할 수 있는 측정치를 제공합니다. 를 통해 이러한 측정치에 액세스할 수 있습니다. 에 대한 자세한 내용은 [설명서](#)를 참조하십시오.

호스트 수준 측정치(예: CPU 사용) 및 캐시 엔진 소프트웨어별 측정치(예: 캐시가 얻은 것과 잃은 것) 모두를 제공합니다. 이러한 메트릭은 60초 간격으로 각 캐시 노드에 대해 측정되어 게시됩니다.

### Important

특정 키 측정치에 경보를 설정하면 캐시 클러스터의 성능이 나빠지기 시작하면 알림을 받을 수 있습니다. 자세한 내용은 이 가이드의 [어떤 메트릭을 모니터링해야 합니까? \(p. 142\)](#)을 참조하십시오.

## 주제

- [호스트 수준 지표 \(p. 139\)](#)
- [Memcached 지표 \(p. 140\)](#)
- [어떤 메트릭을 모니터링해야 합니까? \(p. 142\)](#)
- [메트릭 통계 및 기간 선택 \(p. 143\)](#)
- [Monitoring CloudWatch Cluster and Node Metrics \(p. 143\)](#)

## 호스트 수준 지표

AWS/ElastiCache 네임스페이스에는 다음과 같이 개별 캐시 노드에 대한 호스트 수준 지표가 포함되어 있습니다.

### 참고 항목

- [Memcached 지표 \(p. 140\)](#)

지표	설명	단위
CPUtilization	CPU 사용 백분율.	%
FreeableMemory	호스트에서 사용 가능한 메모리의 양.	바이트
NetworkBytesIn	호스트가 네트워크에서 읽어온 바이트 수	바이트

지표	설명	단위
NetworkBytesOut	호스트가 네트워크에 쓴 바이트 수.	바이트
NetworkPacketsIn	인스턴스가 모든 네트워크 인터페이스에서 받은 패킷 수입니다. 이 지표는 단일 인스턴스에서 수신 트래픽의 볼륨을 패킷 수 기준으로 식별합니다.	개수
NetworkPacketsOut	인스턴스가 모든 네트워크 인터페이스에서 보낸 패킷 수입니다. 이 지표는 단일 인스턴스에서 발신 트래픽의 볼륨을 패킷 수 기준으로 식별합니다.	개수
SwapUsage	호스트에서 사용되는 스왑의 양.	개수

## Memcached 지표

AWS/ElastiCache 네임스페이스에는 다음 Memcached 지표가 포함되어 있습니다.

AWS/ElastiCache 네임스페이스에는 다음과 같이 Memcached stats 명령에서 파생되는 지표가 포함되어 있습니다. 각 지표는 캐시 노드 수준에서 계산됩니다.

Memcached info 명령의 전체 설명서는 [단원을 참조하십시오](#).

### 참고 항목

- [호스트 수준 지표 \(p. 139\)](#)

지표	설명	단위
BytesReadIntoMemcached	캐시 노드가 네트워크에서 읽어온 바이트 수	바이트
BytesUsedForCache	캐시 항목을 저장하는 데 사용된 바이트 수	바이트
BytesWrittenOutFromMemcached	캐시 노드가 네트워크로 작성한 바이트 수	바이트
CasBadval	CAS(Check And Set) 값이 저장된 CAS 값과 일치하지 않을 때 캐시가 수신한 CAS 요청 수	개수
CasHits	요청한 키를 찾았고, CAS 값이 일치할 때 캐시가 수신한 CAS 요청 수	개수
CasMisses	요청한 키를 찾지 못했을 때 캐시가 수신한 CAS 요청 수	개수
CmdFlush	캐시가 수신한 flush 명령 수	개수
CmdGets	캐시가 수신한 get 명령 수	개수
CmdSet	캐시가 수신한 set 명령 수	개수
CurrConnections	동시에 캐시에 연결된 연결 수 집계. ElastiCache는 2~3개의 연결을 사용하여 클러스터를 모니터링합니다.  위의 내용 외에도 memcached는 노드 유형에 사용되는 스레드의 두 배와 동일한 수의 내부 연결을 만듭니다. 다양한 노드 유형에 대한 스레드 수는 <a href="#">해</a>	개수

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
Memcached 지표

지표	설명	단위
	당하는 파라미터 그룹의 <code>Nodetype Specific Parameters</code> 에서 확인할 수 있습니다.  총 연결 수는 클라이언트 연결, 모니터링을 위한 연결 및 위에 언급된 내부 연결의 합계입니다.	
<code>CurrItems</code>	현재 캐시에 저장된 항목 수 집계	개수
<code>DecrHits</code>	요청한 키를 찾았을 때 캐시가 수신한 <code>decrement</code> 요청 수	개수
<code>DecrMisses</code>	요청한 키를 찾지 못했을 때 캐시가 수신한 <code>decrement</code> 요청 수	개수
<code>DeleteHits</code>	요청한 키를 찾았을 때 캐시가 수신한 <code>delete</code> 요청 수	개수
<code>DeleteMisses</code>	요청한 키를 찾지 못했을 때 캐시가 수신한 <code>delete</code> 요청 수	개수
<code>Evictions</code>	새로운 쓰기 공간 확보를 위해 캐시가 제거한 만료 이전 항목 수	개수
<code>GetHits</code>	요청한 키를 찾았을 때 캐시가 수신한 <code>get</code> 요청 수	개수
<code>GetMisses</code>	요청한 키를 찾지 못했을 때 캐시가 수신한 <code>get</code> 요청 수	개수
<code>IncrHits</code>	요청한 키를 찾았을 때 캐시가 수신한 <code>increment</code> 요청 수	개수
<code>IncrMisses</code>	요청한 키를 찾지 못했을 때 캐시가 수신한 <code>increment</code> 요청 수	개수
<code>Reclaimed</code>	새로운 쓰기 공간 확보를 위해 캐시가 제거한 만료 항목 수	개수

Memcached 1.4.14에서는 다음과 같은 지표가 추가 제공됩니다.

지표	설명	단위
<code>BytesUsedForHash</code>	현재 해시 테이블에서 사용 중인 바이트 수	바이트
<code>CmdConfigGet</code>	누적된 <code>config get</code> 요청 수	개수
<code>CmdConfigSet</code>	누적된 <code>config set</code> 요청 수	개수
<code>CmdTouch</code>	누적된 <code>touch</code> 요청 수	개수
<code>CurrConfig</code>	현재 저장된 구성 수	개수
<code>EvictedUnfetched</code>	LRU(Least Recently Used) 캐시에서 설정 이후 한 번도 사용하지 않아서 제거된 유효 항목 수	개수
<code>ExpiredUnfetched</code>	LRU에서 설정 이후 한 번도 사용하지 않아서 다시 회수된 만료 항목 수	개수
<code>SlabsMoved</code>	이동한 슬래브 페이지 총 수	개수

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
어떤 메트릭을 모니터링해야 하나요?

지표	설명	단위
TouchHits	새로운 만료 시간 지정 이후 사용한 적이 있는 키 수	개수
TouchMisses	사용한 적은 있지만 찾을 수 없는 항목 수	개수

AWS/ElastiCache 네임스페이스에는 다음과 같이 계산된 캐시 수준 지표가 포함되어 있습니다.

지표	설명	단위
NewConnections	캐시가 수신한 새로운 연결 수. 이는 일정 기간 동안 total_connections의 변화를 기록하여 memcached total_connections 통계로부터 파생됩니다. ElastiCache에 예약되는 연결로 인해 항상 1 이상의 값을 갖습니다.	개수
NewItems	캐시에 저장된 새로운 항목 수. 이는 일정 기간 동안 total_items의 변화를 기록하여 memcached total_items 통계로부터 파생됩니다.	개수
UnusedMemory	데이터에서 사용하지 않는 메모리 크기. 이는 limit_maxbytes에서 바이트를 빼 Memcached 통계 limit_maxbytes 및 바이트로부터 파생됩니다.  메모리는 데이터 뿐만 아니라 Memcached 오버헤드에서도 사용되기 때문에 UnusedMemory를 데이터 추가 시 사용할 수 있는 메모리 양으로 간주해서는 안 됩니다. 아직 미사용 메모리가 있는데도 불구하고 데이터를 제거해야 하는 경우가 발생할 수도 있습니다.  자세한 내용은 <a href="#">Memcached 항목 메모리 사용 단원</a> 을 참조하십시오.	바이트

## 어떤 메트릭을 모니터링해야 하나요?

다음 측정치는 성능에 대한 좋은 정보를 제공합니다. 대부분의 경우 이러한 측정치에 대해 성능 문제가 생기기 전에 수정 조치를 취할 수 있도록 경보를 설정하는 것이 좋습니다.

모니터링할 측정치

- [CPUUtilization \(p. 142\)](#)
- [SwapUsage \(p. 143\)](#)
- [Evictions \(p. 143\)](#)
- [CurrConnections \(p. 143\)](#)

### CPUUtilization

이는 비율(%)로 보고된 호스트 수준 측정치입니다. 자세한 내용은 [호스트 수준 지표 \(p. 139\)](#) 단원을 참조하십시오.

Memcached는 다중 스레드이므로 이 측정치가 90%에 이를 수 있습니다. 이 임계값을 초과할 경우 대형 캐시 노드 유형을 사용하여 캐시 클러스터를 스케일 업하거나 더 많은 캐시 노드를 추가하여 스케일 아웃합니다.

## SwapUsage

이는 바이트로 보고된 호스트 수준 측정치입니다. 자세한 내용은 [호스트 수준 지표 \(p. 139\)](#) 단원을 참조하십시오.

이 메트릭은 50MB를 초과하지 않아야 합니다. 초과하면 ConnectionOverhead 매개 변수 값을 높이는 것이 좋습니다.

## Evictions

이것은 캐시 엔진 메트릭입니다. 애플리케이션 요구 사항에 따라 이 메트릭에 대한 경보 임계값을 결정하는 것이 좋습니다.

선택한 임계값을 초과할 경우 대형 노드 유형을 사용하여 클러스터를 스케일 업하거나 더 많은 노드를 추가하여 스케일 아웃합니다.

## CurrConnections

이것은 캐시 엔진 메트릭입니다. 애플리케이션 요구 사항에 따라 이 메트릭에 대한 경보 임계값을 결정하는 것이 좋습니다.

CurrConnections 개수가 증가하는 것은 애플리케이션에 문제가 있음을 나타내며 이 문제를 해결하려면 애플리케이션 동작을 확인해야 합니다.

## 메트릭 통계 및 기간 선택

에서 각 측정치의 통계 및 기간을 선택하도록 허용하며 모든 조합이 유용한 것은 아닙니다. 예를 들어, CPUUtilization에 대한 Average, Minimum 및 Maximum 통계는 유용하지만 Sum 통계는 유용하지 않습니다.

모든 샘플은 각 개별 캐시 노드에 대해 60초 동안 게시됩니다. 60초 기간 동안 캐시 노드 메트릭은 단일 샘플만 포함할 수 있습니다.

캐시 노드 메트릭을 검색하는 방법에 대한 자세한 내용은 [Monitoring CloudWatch Cluster and Node Metrics \(p. 143\)](#)를 참조하십시오.

## Monitoring CloudWatch Cluster and Node Metrics

ElastiCache와 CloudWatch가 서로 통합되어 있어서 다양한 지표를 수집할 수 있습니다. CloudWatch를 사용하여 이러한 지표를 모니터링할 수 있습니다.

### Note

다음 예제를 실행하려면 CloudWatch 명령줄 도구가 필요합니다. CloudWatch에 대해 자세한 내용을 알아보고 개발자 도구를 다운로드하려면 [CloudWatch제품 페이지](#)를 참조하십시오.

다음 절차에서는 CloudWatch를 사용하여 지난 시간 캐시 클러스터의 스토리지 공간 통계를 수집하는 방법을 보여줍니다.

### Note

아래 예제에 나온 startTime 및 endTime 값은 설명을 돕기 위해 지정되었습니다. 따라서 캐시 노드의 올바른 시작 및 종료 시간 값으로 대체해야 합니다.

ElastiCache 제한에 대한 정보는 ElastiCache의 [AWS 서비스 제한](#)을 참조하십시오.

## Monitoring CloudWatch Cluster and Node Metrics (콘솔)

캐시 클러스터의 CPU 사용률 통계를 수집하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.

- 지표를 확인할 캐시 노드를 선택합니다.

#### Note

20개보다 많은 노드를 선택하면 콘솔에 지표가 표시되지 않습니다.

- AWS Management Console의 [Cache Clusters] 페이지에서 하나 이상의 캐시 클러스터 이름을 클릭합니다.

캐시 클러스터의 세부 정보 페이지가 나타납니다.

- 창 맨 위의 [Nodes] 탭을 클릭합니다.
- 세부 정보 창의 [Nodes] 탭에서 지표를 확인할 캐시 노드를 선택합니다.

사용 가능한 CloudWatch 지표 목록이 콘솔 창 하단에 나타납니다.

- [CPU Utilization] 지표를 클릭합니다.

선택한 지표가 표시된 CloudWatch 콘솔이 열립니다. [Statistic] 및 [Period] 드롭다운 목록 상자와 [Time Range] 탭을 사용하여 표시되는 지표를 변경할 수 있습니다.

## Monitoring CloudWatch Cluster and Node Metrics CloudWatch CLI 사용

캐시 클러스터의 CPU 사용률 통계를 수집하려면

- Linux, macOS, or Unix의 경우:

```
aws cloudwatch get-metric-statistics \
 --namespace AWS/ElastiCache \
 --metric-name CPUUtilization \
 --dimensions='[{"Name":"CacheClusterId","Value":"mycacheclass"},
{"Name":"CacheNodeId","Value":"0002"}]' \
 --statistics=Average \
 --start-time 2018-07-05T00:00:00 \
 --end-time 2018-07-06T00:00:00 \
 --period=3600
```

Windows의 경우:

```
aws cloudwatch get-metric-statistics ^
 --namespace AWS/ElastiCache ^
 --metric-name CPUUtilization ^
 --dimensions='[{"Name":"CacheClusterId","Value":"mycacheclass"},
{"Name":"CacheNodeId","Value":"0002"}]' ^
 --statistics=Average ^
 --start-time 2018-07-05T00:00:00 ^
 --end-time 2018-07-06T00:00:00 ^
 --period=3600
```

## Monitoring CloudWatch Cluster and Node Metrics CloudWatch API 사용

캐시 클러스터의 CPU 사용률 통계를 수집하려면

- CloudWatch API GetMetricStatistics를 다음 파라미터와 함께 호출합니다. 시작 및 종료 시간은 예제와 같이 표시되며 적절한 서각 및 종결 서각을 대체해야 합니다.

API 버전 2015-02-02

- `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=mycachecuster,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=mycachecuster"
&Dimensions.member.2="CacheNodeId=0002"
&Namespace=AWS/ElastiCache
&MeasureName=CPUUtilization
&Timestamp=2018-07-07T17%3A48%3A21.746Z
&AWSAccessKeyId=<AWS Access Key ID>
&Signature=<Signature>
```

## ElastiCache 이벤트 모니터링

클러스터에서 중요 이벤트가 발생하면 ElastiCache는 특정 Amazon SNS 주제에 알림을 전송합니다. 이러한 예에는 노드 추가 실패, 노드 추가 성공, 보안 그룹 수정 등이 있습니다. 주요 이벤트를 모니터링하면 클러스터의 현재 상태를 파악할 수 있으며, 이벤트에 따라 교정 작업을 수행할 수도 있습니다.

### 주제

- [ElastiCache Amazon SNS 알림 관리 \(p. 145\)](#)
- [ElastiCache 이벤트 보기 \(p. 148\)](#)
- [이벤트 알림 및 Amazon SNS \(p. 150\)](#)

## ElastiCache Amazon SNS 알림 관리

Amazon Simple Notification Service(Amazon SNS)를 사용하여 중요한 클러스터 이벤트에 대해 알림을 보내도록 ElastiCache를 구성할 수 있습니다. 이러한 예에서는 Amazon SNS 항목의 ARN(Amazon 리소스 이름)으로 클러스터를 구성하여 알림을 받습니다.

### Note

이 항목에서는 Amazon SNS에 가입했으며 Amazon SNS 주제를 설정 및 구독했다고 가정합니다. 이렇게 하는 방법에 대한 정보는 [Amazon Simple Notification Service 개발자 안내서](#) 단원을 참조하십시오.

## Amazon SNS 주제 추가

다음 단원은 AWS 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 Amazon SNS 주제를 추가하는 방법을 보여줍니다.

### Amazon SNS 주제 (콘솔) 추가

다음 절차는 클러스터에 대해 Amazon SNS 주제를 추가하는 방법을 보여줍니다.

#### Note

이 프로세스는 Amazon SNS 주제를 수정하는 데에도 사용할 수 있습니다.

(콘솔) 클러스터에 대해 Amazon SNS 주제를 추가 또는 수정하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 클러스터에서 Amazon SNS 주제 ARN을 추가 또는 수정할 클러스터를 선택합니다.
3. [Modify]를 선택합니다.
4. SNS 알림 주제 아래의 클러스터 수정에서 추가할 SNS 주제를 선택하거나 수동 ARN 입력을 선택하고 Amazon SNS 주제의 ARN을 입력합니다.
5. [Modify]를 선택합니다.

### Amazon SNS 주제 (AWS CLI) 추가

클러스터에 대해 Amazon SNS 주제를 추가 또는 수정하려면 AWS CLI 명령 `modify-cache-cluster`를 사용합니다.

다음 코드 예제는 Amazon SNS 주제를 `my-cluster`에 추가합니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-cluster \
 --notification-topic-arn arn:aws:sns:us-west-2:565419523791:ElastiCacheNotifications
```

Windows의 경우:

```
aws elasticache modify-cache-cluster ^
 --cache-cluster-id my-cluster ^
 --notification-topic-arn arn:aws:sns:us-west-2:565419523791:ElastiCacheNotifications
```

자세한 내용은 `modify-cache-cluster`를 참조하십시오.

### Amazon SNS 주제 (ElastiCache API) 추가

클러스터에 대해 Amazon SNS 주제를 추가 또는 수정하려면 다음 파라미터와 함께 `ModifyCacheCluster` 작업을 호출합니다.

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

#### Example

```
https://elasticache.amazonaws.com/
```

```
?Action=ModifyCacheCluster
&ApplyImmediately=false
&CacheClusterId=my-cluster
&NotificationTopicArn=arn%3Aaws%3Asns%3Aus-
west-2%3A565419523791%3AElastiCacheNotifications
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

자세한 내용은 [ModifyCacheCluster](#)를 참조하십시오.

## Amazon SNS 알림 활성화 및 비활성화

클러스터에 대해 알림을 켜거나 끌 수 있습니다. 다음 절차는 Amazon SNS 알림을 비활성화하는 방법을 보여줍니다.

### Amazon SNS 알림 (콘솔) 활성화 및 비활성화

AWS Management 콘솔을 사용하여 Amazon SNS 알림을 비활성화하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. Memcached를 실행 중인 클러스터의 목록을 보려면 탐색 창에서 Memcached를 선택합니다.
3. 알림을 수정할 클러스터의 이름 왼쪽에 있는 확인란을 선택합니다.
4. [Modify]를 선택합니다.
5. [Topic for SNS Notification] 아래의 [Modify Cluster]에서 [Disable Notifications]를 선택합니다.
6. [Modify]를 선택합니다.

### Amazon SNS 알림 (AWS CLI) 활성화 및 비활성화

Amazon SNS 알림을 비활성화하려면 다음 파라미터와 함께 `modify-cache-cluster` 명령을 사용합니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache modify-cache-cluster \
 --cache-cluster-id my-cluster \
 --notification-topic-status inactive
```

Windows의 경우:

```
aws elasticache modify-cache-cluster ^
 --cache-cluster-id my-cluster ^
 --notification-topic-status inactive
```

### Amazon SNS 알림(ElastiCache API) 활성화 및 비활성화

Amazon SNS 알림을 비활성화하려면 다음 파라미터와 함께 `ModifyCacheCluster` 작업을 호출합니다.

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

이 호출은 다음과 비슷한 출력을 반환합니다.

#### 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=AWS4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

## ElastiCache 이벤트 보기

ElastiCache는 클러스터 인스턴스, 보안 그룹 및 파라미터 그룹과 관련된 이벤트를 기록합니다. 여기에는 이벤트 날짜 및 시간, 이벤트의 원본 이름 및 원본 유형, 이벤트 설명 등의 정보가 포함됩니다. ElastiCache 콘솔, AWS CLI `describe-events` 명령 또는 ElastiCache API 작업 `DescribeEvents`을 사용하여 로그에서 이벤트를 쉽게 검색할 수 있습니다.

다음 절차는 지난 24시간(1440분) 동안의 모든 ElastiCache 이벤트를 보는 방법을 보여줍니다.

## ElastiCache 이벤트 (콘솔) 보기

다음 절차는 ElastiCache 콘솔을 사용하여 이벤트를 표시합니다.

ElastiCache 콘솔을 사용하여 이벤트를 보려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 사용 가능한 모든 이벤트의 목록을 보려면 탐색 창에서 이벤트를 선택합니다.

이벤트 화면에서 목록의 각 행은 이벤트 하나를 나타내며 이벤트 소스, 이벤트 유형(cache-cluster, cache-parameter-group, cache-security-group 또는 cache-subnet-group), 이벤트의 GMT 시간 및 이벤트 설명을 표시합니다.

[Filter]를 사용하여 이벤트 목록에서 모든 이벤트를 볼지 특정 유형의 이벤트만 볼지를 지정할 수 있습니다.

## ElastiCache 이벤트 (AWS CLI) 보기

ElastiCache를 사용하여 AWS CLI 이벤트의 목록을 생성하려면 `describe-events` 명령을 사용합니다. 선택적 파라미터를 사용하여 나열된 이벤트의 유형, 나열된 이벤트의 기간, 나열할 이벤트의 최대 수 등을 제어할 수 있습니다.

다음 코드는 최대 40개의 캐시 클러스터 이벤트를 나열합니다.

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

다음 코드는 지난 24시간(1440분) 동안의 모든 이벤트를 나열합니다.

```
aws elasticache describe-events --duration 1440
```

describe-events 명령의 출력은 다음과 같습니다.

```
{
 "Events": [
 {
 "Date": "2017-03-29T22:17:37.781Z",
 "Message": "Added cache node 0001 in Availability Zone us-west-2a",
 "SourceIdentifier": "mem01",
 "SourceType": "cache-cluster"
 },
 {
 "Date": "2017-03-29T22:17:37.769Z",
 "Message": "Cache cluster created",
 "SourceIdentifier": "mem01",
 "SourceType": "cache-cluster"
 }
]
}
```

사용 가능한 파라미터 및 허용된 파라미터 값과 같은 자세한 내용은 [describe-events](#)를 참조하십시오.

## ElastiCache 이벤트 (ElastiCache API) 보기

ElastiCache API를 사용하여 ElastiCache 이벤트의 목록을 생성하려면 DescribeEvents 작업을 사용합니다. 선택적 파라미터를 사용하여 나열된 이벤트의 유형, 나열된 이벤트의 기간, 나열할 이벤트의 최대 수 등을 제어할 수 있습니다.

다음 코드는 40개의 최신 cache-cluster 이벤트를 나열합니다.

```
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>
```

다음 코드는 지난 24시간(1440분) 동안의 cache-cluster 이벤트를 나열합니다.

```
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>
```

위 작업을 통해 다음과 비슷한 출력이 생성되어야 합니다.

```
<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>
```

사용 가능한 파라미터 및 허용된 파라미터 값과 같은 자세한 내용은 [DescribeEvents](#)를 참조하십시오.

## 이벤트 알림 및 Amazon SNS

ElastiCache는 캐시 클러스터에서 중요 이벤트가 발생하는 경우 Amazon Simple Notification Service(SNS)를 사용하여 메시지를 게시할 수 있습니다. 이 기능은 캐시 클러스터의 개별 캐시 노드 Endpoint에 연결된 클라이언트 머신의 서버 목록을 새로 고침하는 데 사용될 수 있습니다.

### Note

이용 요금 정보 및 Amazon SNS 설명서 링크 등 Amazon Simple Notification Service(SNS)에 대한 자세한 내용은 [Amazon SNS 제품 페이지](#)를 참조하십시오.

알림은 지정된 Amazon SNS 주제에 대해 게시됩니다. 다음은 알림에 대한 요구 사항입니다.

- ElastiCache 알림에 대해 주제 하나만 구성할 수 있습니다.
- Amazon SNS 주제를 소유한 AWS 계정은 알림이 활성화된 캐시 클러스터를 소유하는 계정과 동일해야 합니다.

## 예제 ElastiCache SNS 알림

다음은 캐시 클러스터 생성 성공에 대한 ElastiCache Amazon SNS 알림을 보여주는 예입니다.

### Example

```
{
 "Date": "2015-12-05T01:02:18.336Z",
 "Message": "Cache cluster created",
 "SourceIdentifier": "memcache-ni",
 "SourceType": "cache-cluster"
}
```

## ElastiCache 이벤트

다음 ElastiCache 이벤트는 Amazon SNS 알림을 트리거합니다.

이벤트 이름	메시지	설명
ElastiCache:AddCacheNodeComplete	<code>cluster-name</code>	캐시 노드가 캐시 클러스터에 추가되었고 사용할 준비가 되어 있습니다.

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
이벤트 알림 및 Amazon SNS

이벤트 이름	메시지	설명
ElastiCache 무료 IP 주소가 부족함으로 인한 :AddCacheNodeFailed	<i>cluster-name</i>	사용 가능한 IP 주소가 충분하지 않아 캐시 노드를 추가하지 못했습니다.
ElastiCache:CacheClusterParametersChanged	<i>cluster-name</i>	하나 이상의 캐시 클러스터 매개변수가 변경되었습니다.
ElastiCache:CacheClusterProvisioningComplete	<i>cluster-name</i>	캐시 클러스터의 프로비저닝이 완료되어 캐시 클러스터에 있는 캐시 노드를 사용할 수 있습니다.
ElastiCache 호환되지 않는 네트워크 상태로 인한 :CacheClusterProvisioningFailed	<i>cluster-name</i>	존재하지 않는 Virtual Private Cloud(VPC)에서 새로운 캐시 클러스터를 실행하려고 시도했습니다.
ElastiCache:CacheClusterScalingComplete	<i>cluster-name</i>	: 캐시 클러스터의 스케일 업이 성공적으로 완료되었습니다.
ElastiCache:CacheClusterScalingFailed	<i>cluster-name</i>	캐시 클러스터에 대한 스케일 업 작업이 실패했습니다.
ElastiCache:CacheClusterSecurityGroupModified	<i>cluster-name</i>	다음 이벤트 중 하나가 발생했습니다. <ul style="list-style-type: none"> <li>캐시 클러스터를 위한 승인된 캐시 보안 그룹이 수정되었습니다.</li> <li>하나 이상의 새로운 EC2 보안 그룹이 캐시 클러스터와 연결된 캐시 보안 그룹 중 하나에서 승인되었습니다.</li> <li>하나 이상의 EC2 보안 그룹이 캐시 클러스터와 연결된 캐시 보안 그룹 중 하나에서 승인이 취소되었습니다.</li> </ul>

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
이벤트 알림 및 Amazon SNS

이벤트 이름	메시지	설명
ElastiCache:CacheNodeReplaceStarted	<code>cluster-name</code>	<p>ElastiCache가 캐시 노드를 실행하는 호스트 성능이 저하되었거나 연결되지 않음을 감지하여 캐시 노드 교체를 시작했습니다.</p> <p><b>Note</b></p> <p>교체된 캐시 노드의 DNS 항목은 변경되지 않습니다.</p> <p>대부분의 경우에 이 이벤트가 발생할 때 클라이언트의 서버 목록을 새로 고침하지 않아도 됩니다. 하지만 일부 캐시 클라이언트 라이브러리는 ElastiCache가 캐시 노드를 교체한 후에도 캐시 노드 사용을 중단할 수 있습니다. 이 경우에 애플리케이션은 이 이벤트가 발생할 때 서버 목록을 새로 고침해야 합니다.</p>
ElastiCache:CacheNodeReplaceCompleted	<code>cluster-name</code>	<p>ElastiCache가 캐시 노드를 실행하는 호스트 성능이 저하되었거나 연결되지 않음을 감지하여 캐시 노드 교체를 완료했습니다.</p> <p><b>Note</b></p> <p>교체된 캐시 노드의 DNS 항목은 변경되지 않습니다.</p> <p>대부분의 경우에 이 이벤트가 발생할 때 클라이언트의 서버 목록을 새로 고침하지 않아도 됩니다. 하지만 일부 캐시 클라이언트 라이브러리는 ElastiCache가 캐시 노드를 교체한 후에도 캐시 노드 사용을 중단할 수 있습니다. 이 경우에 애플리케이션은 이 이벤트가 발생할 때 서버 목록을 새로 고침해야 합니다.</p>
ElastiCache:CacheNodesRebooted	<code>cluster-name</code>	<p>하나 이상의 캐시 노드가 재부팅되었습니다.</p> <p>메시지(Memcached): "Cache node %s shutdown", 두 번째 메시지: "Cache node %s restarted"</p>
ElastiCache>DeleteCacheClusterCompleted	<code>cluster-name</code>	<p>캐시 클러스터 및 연결된 모든 캐시 노드 삭제를 완료했습니다.</p>
ElastiCache:FailoverComplete(Redis 전용)	<code>cluster-name</code>	<p>복제본 노드에 대한 장애 조치가 성공했습니다.</p>

이벤트 이름	메시지	설명
ElastiCache:NodeReplacementCancelled	<code>cluster-name</code>	교체가 예약되어 있는 클러스터의 노드가 더 이상 교체 예약이 되지 않습니다.
ElastiCache:NodeReplacementRescheduled	<code>cluster-name</code>	이전에 교체가 예약되어 있는 클러스터의 노드가 알림에 설명된 새 기간 동안 교체가 예약됩니다.  수행할 수 있는 작업에 대한 자세한 내용은 <a href="#">노드 교체 (p. 42)</a> 단원을 참조하십시오.
ElastiCache:NodeReplacementScheduled	<code>cluster-name</code>	클러스터의 노드가 알림에 설명된 기간 동안 교체가 예약됩니다.  수행할 수 있는 작업에 대한 자세한 내용은 <a href="#">노드 교체 (p. 42)</a> 단원을 참조하십시오.
ElastiCache:RemoveCacheNodeComplete	<code>cluster-name</code>	캐시 노드가 캐시 클러스터에서 제거되었습니다.
ElastiCache:SnapshotComplete(Redis 전용)	<code>cluster-name</code>	캐시 스냅샷이 성공적으로 완료되었습니다.
ElastiCache:SnapshotFailed(Redis 전용)	<code>cluster-name</code>	캐시 스냅샷이 실패했습니다. 원인에 대한 자세한 내용은 클러스터의 캐시 이벤트를 참조하십시오.  스냅샷을 설명할 경우 <a href="#">DescribeSnapshots</a> 를 참조하십시오. 상태는 <code>failed</code> 입니다.

## 관련 항목

- [ElastiCache 이벤트 보기 \(p. 148\)](#)

# 비용 할당 태그를 사용한 비용 모니터링

Amazon ElastiCache에서 리소스에 비용 할당 태그를 추가하면 인보이스 비용을 리소스 태그 값으로 그룹화하여 비용을 추적할 수 있습니다.

ElastiCache 비용 할당 태그는 사용자가 정의하고 ElastiCache 리소스에 연결하는 키 값 페어입니다. 키와 값은 대/소문자를 구분합니다. 태그 키를 사용하여 범주를 정의할 수 있으며 태그 값은 해당 범주의 항목일 수 있습니다. 예를 들어, `CostCenter`의 태그 키와 `10010`의 태그 값을 정의하여 리소스가 10010 코스트 센터에 할당됨을 나타냅니다. `Environment`와 같은 키 및 `test` 또는 `production`과 같은 값을 사용하여 태그로 리소스를 테스트나 프로덕션에 사용되는 것으로 지정할 수도 있습니다. 리소스 관련 비용을 보다 쉽게 추적할 수 있도록 하기 위해 일관된 태그 키 세트를 사용하는 것이 좋습니다.

비용 할당 태그를 사용하여 자신만의 비용 구조를 반영하도록 AWS 청구서를 구성합니다. 이렇게 하려면 가입하여 태그 키 값이 포함된 AWS 계정 청구서를 가져옵니다. 그런 다음 같은 태그 키 값을 가진 리소스에 따라 결제 정보를 구성하여 리소스 비용의 합을 볼 수 있습니다. 예를 들어, 특정 애플리케이션 이름으로 여러

리소스에 태그를 지정한 다음 결제 정보를 구성하여 여러 서비스에 걸친 해당 애플리케이션의 총 비용을 볼 수 있습니다.

또한 태그를 결합하여 보다 세부적인 수준으로 비용을 추적할 수 있습니다. 예를 들어, 리전별 서비스 비용을 추적하려면 Service 및 Region 태그 키를 사용할 수 있습니다. 하나의 리소스에서 ElastiCache 및 ###(###) 값이 있을 수 있으며 다른 리소스에서 ElastiCache 및 ##(#####) 값이 있을 수 있습니다. 리전별로 구분된 총 ElastiCache 비용을 볼 수 있습니다. 자세한 내용은 AWS Billing and Cost Management 사용 설명서의 [비용 할당 태그 사용](#)을 참조하십시오.

Memcached 클러스터에 ElastiCache 비용 할당 태그를 추가할 수 있습니다. 태그를 추가, 나열, 수정, 복사 또는 제거할 때 이 작업은 지정된 클러스터에만 적용됩니다.

### ElastiCache 비용 할당 태그 특성

- 비용 할당 태그는 CLI 및 API 작업에서 ARN으로 지정된 ElastiCache 리소스에 적용됩니다. 리소스 유형은 "클러스터"입니다.

샘플 ARN: `arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Memcached: 태그가 클러스터에만 적용됩니다.

샘플 arn: `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- 태그 키는 태그의 필수 이름입니다. 키의 문자열 값은 1~128자(유니코드 문자) 사이가 될 수 있으며 aws:로 시작할 수 없습니다. 문자열에는 유니코드 문자, 숫자, 공백, 밑줄(\_), 마침표(.), 콜론(:), 백슬래시(\), 등호(=), 더하기 기호(+), 하이픈(-), at 기호(@) 집합만 포함될 수 있습니다.
- 태그 값은 태그의 선택적 값입니다. 값의 문자열 값은 1~256자(유니코드 문자) 사이가 될 수 있으며 aws:로 시작할 수 없습니다. 문자열에는 유니코드 문자, 숫자, 공백, 밑줄(\_), 마침표(.), 콜론(:), 백슬래시(\), 등호(=), 더하기 기호(+), 하이픈(-), at 기호(@) 집합만 포함될 수 있습니다.
- ElastiCache 리소스는 최대 50개의 태그를 보유할 수 있습니다.
- 태그 세트의 값이 고유하지 않습니다. 예를 들어, 두 키 Service와 Application에 ElastiCache 값이 있는 태그 세트가 있을 수 있습니다.

AWS는 태그에 의미론적 의미를 적용하지 않습니다. 태그는 엄격히 문자열로 해석됩니다. AWS에서는 ElastiCache 리소스에 어떠한 태그도 자동으로 설정하지 않습니다.

ElastiCache 관리 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 ElastiCache 리소스의 태그를 추가, 나열, 수정 또는 제거할 수 있습니다.

### 주제

- [ElastiCache 콘솔을 사용한 태그 관리 \(p. 154\)](#)
- [AWS CLI를 사용하여 비용 할당 태그 관리 \(p. 155\)](#)
- [ElastiCache API를 사용하여 비용 할당 태그 관리 \(p. 158\)](#)

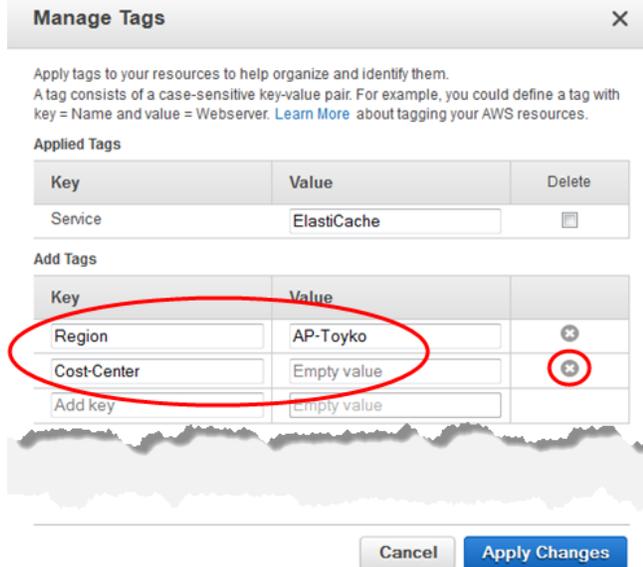
## ElastiCache 콘솔을 사용한 태그 관리

Amazon ElastiCache 콘솔을 사용하여 비용 할당 태그를 추가, 수정 또는 제거할 수 있습니다.

다음 절차에서는 ElastiCache 관리 콘솔을 사용하여 하나 이상의 비용 할당 태그를 보거나 추가, 수정 또는 삭제하는 과정을 안내합니다.

AWS Management 콘솔을 사용하여 Memcached 클러스터에서 태그 관리

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. Memcached를 선택합니다.
3. 태그를 추가할 클러스터 이름 왼쪽에 있는 확인란을 선택합니다.
4. [Manage Tags]를 선택한 다음 대화 상자를 사용하여 태그를 관리합니다.



5. 추가, 수정 또는 제거하려는 각 태그에 대해 다음을 수행합니다.

태그를 추가, 수정 또는 제거하려면

- 태그를 추가하려면 [Key] 열의 [Add key]가 표시된 상자에 키 이름을 입력하고 키 이름 오른쪽에 있는 상자에 선택적 값을 입력합니다.
- 태그를 수정하려면 [Value] 열에 새 값을 입력하거나 태그의 기존 값을 제거합니다.
- 태그를 제거하려면 태그 오른쪽에 있는 [X]를 선택합니다.

6. 완료되면 [Apply Changes]를 선택합니다.

## AWS CLI를 사용하여 비용 할당 태그 관리

AWS CLI를 사용하여 비용 할당 태그를 추가, 수정 또는 제거할 수 있습니다.

비용 할당 태그는 ElastiCache for Memcached 클러스터에 적용됩니다. 태그를 지정할 클러스터는 Amazon 리소스 이름(ARN)을 사용해 지정합니다.

샘플 `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

샘플 `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

주제

- [AWS CLI를 사용하여 태그 나열 \(p. 156\)](#)
- [AWS CLI를 사용하여 태그 추가 \(p. 156\)](#)
- [AWS CLI를 사용하여 태그 수정 \(p. 157\)](#)
- [AWS CLI를 사용하여 태그 제거 \(p. 157\)](#)

## AWS CLI를 사용하여 태그 나열

`list-tags-for-resource` 작업을 사용하여 기존 ElastiCache 리소스에서 태그를 나열하는 데 AWS CLI를 사용할 수 있습니다.

다음 코드는 AWS CLI를 사용하여 us-west-2 리전에 있는 Memcached 클러스터 `my-cluster`의 태그를 나열합니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache list-tags-for-resource \
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

Windows의 경우:

```
aws elasticache list-tags-for-resource ^\
 --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

이 작업의 출력은 다음 리소스의 모든 태그 목록과 유사합니다.

```
{
 "TagList": [
 {
 "Value": "10110",
 "Key": "CostCenter"
 },
 {
 "Value": "EC2",
 "Key": "Service"
 }
]
}
```

리소스에 태그가 없으면 출력은 빈 TagList가 됩니다.

```
{
 "TagList": []
}
```

자세한 내용은 ElastiCache에 대한 AWS CLI `list-tags-for-resource`를 참조하십시오.

## AWS CLI를 사용하여 태그 추가

`add-tags-to-resource` CLI 작업을 사용하여 기존 ElastiCache 리소스에 태그를 추가하는 데 AWS CLI를 사용할 수 있습니다. 리소스에 태그 키가 없으면 키와 값이 리소스에 추가됩니다. 리소스에 이미 키가 있는 경우 해당 키와 연결된 값이 새 값으로 업데이트됩니다.

다음 코드에서는 AWS CLI를 사용하여 us-west-2 리전의 `my-cluster` 클러스터에 각각 `elasticache` 및 `us-west-2` 값을 갖는 `Service` 및 `Region` 키를 추가합니다.

Linux, macOS, or Unix의 경우:

```
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
```

Windows의 경우:

```
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
```

이 작업의 출력은 다음 작업 후 리소스의 모든 태그 목록과 유사합니다.

```
{
 "TagList": [
 {
 "Value": "elasticache",
 "Key": "Service"
 },
 {
 "Value": "us-west-2",
 "Key": "Region"
 }
]
}
```

자세한 내용은 ElastiCache에 대한 AWS CLI [add-tags-to-resource](#)를 참조하십시오.

AWS CLI를 통해 [create-cache-cluster](#) 작업을 사용하여 새 클러스터를 생성할 때 클러스터에 태그를 추가할 수도 있습니다. ElastiCache 관리 콘솔을 사용하는 경우 클러스터를 생성할 때 태그를 추가할 수 없습니다. 클러스터가 생성된 후에는 콘솔을 사용하여 클러스터에 태그를 추가할 수 있습니다.

## AWS CLI를 사용하여 태그 수정

AWS CLI를 사용하여 ElastiCache for Memcached 클러스터의 태그를 수정할 수 있습니다.

태그를 수정하려면

- [add-tags-to-resource](#)를 사용하여 새 태그 및 값을 추가하거나 기존 태그에 연결된 값을 변경합니다.
- [remove-tags-from-resource](#)를 사용하여 리소스에서 지정된 태그를 제거합니다.

두 작업 중 하나의 출력은 지정된 클러스터의 태그와 이 태그의 값이 나열된 목록입니다.

## AWS CLI를 사용하여 태그 제거

[remove-tags-from-resource](#) 작업을 사용하여 기존 ElastiCache for Memcached 클러스터에서 태그를 제거하는 데 AWS CLI를 사용할 수 있습니다.

다음 코드에서는 AWS CLI를 사용하여 us-west-2 리전의 my-cluster 클러스터에서 Service 및 Region 키를 가진 태그를 제거합니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache remove-tags-from-resource \
--resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
--tag-keys PM Service
```

Windows의 경우:

```
aws elasticache remove-tags-from-resource ^
```

```
--resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
--tag-keys PM Service
```

이 작업의 출력은 다음 작업 후 리소스의 모든 태그 목록과 유사합니다.

```
{
 "TagList": []
}
```

자세한 내용은 ElastiCache에 대한 AWS CLI [remove-tags-from-resource](#)를 참조하십시오.

## ElastiCache API를 사용하여 비용 할당 태그 관리

ElastiCache API를 사용하여 비용 할당 태그를 추가, 수정 또는 제거할 수 있습니다.

비용 할당 태그는 Memcached용 ElastiCache 클러스터에 적용됩니다. 태그를 지정할 클러스터는 Amazon 리소스 이름(ARN)을 사용해 지정합니다.

샘플 arn: arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster

주제

- [ElastiCache API를 사용하여 태그 나열 \(p. 158\)](#)
- [ElastiCache API를 사용하여 태그 추가 \(p. 158\)](#)
- [ElastiCache API를 사용하여 태그 수정 \(p. 159\)](#)
- [ElastiCache API를 사용하여 태그 제거 \(p. 159\)](#)

## ElastiCache API를 사용하여 태그 나열

ElastiCache API를 통해 [ListTagsForResource](#) 작업을 사용하여 기존 리소스에서 태그를 나열할 수 있습니다.

다음 코드는 ElastiCache API를 사용하여 us-west-2 리전에 있는 리소스 my-cluster의 태그를 나열합니다.

```
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>
```

## ElastiCache API를 사용하여 태그 추가

[AddTagsToResource](#) 작업을 사용하여 기존 ElastiCache 클러스터에 태그를 추가하는 데 ElastiCache API를 사용할 수 있습니다. 리소스에 태그 키가 없으면 키와 값이 리소스에 추가됩니다. 리소스에 이미 키가 있는 경우 해당 키와 연결된 값이 새 값으로 업데이트됩니다.

다음 코드는 ElastiCache API 사용하여 us-west-2 리전의 리소스 my-cluster에 각각 elasticache 및 us-west-2 값을 갖는 Service 및 Region 키를 추가합니다.

```
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>
```

자세한 내용은 Amazon ElastiCache API Reference의 [AddTagsToResource](#)를 참조하십시오.

## ElastiCache API를 사용하여 태그 수정

ElastiCache API를 사용하여 ElastiCache 클러스터의 태그를 수정할 수 있습니다.

태그 값을 수정하려면

- [AddTagsToResource](#) 작업을 사용하여 새 태그와 값을 추가하거나 기존 태그의 값을 변경합니다.
- [RemoveTagsFromResource](#)를 사용하여 리소스에서 태그를 제거합니다.

두 작업 중 하나의 출력은 지정된 리소스의 태그 목록과 값입니다.

[RemoveTagsFromResource](#)를 사용하여 리소스에서 태그를 제거합니다.

## ElastiCache API를 사용하여 태그 제거

[RemoveTagsFromResource](#) 작업을 사용하여 기존 ElastiCache for Memcached 클러스터에서 태그를 제거하는 데 ElastiCache API를 사용할 수 있습니다.

다음 코드에서는 ElastiCache API를 사용하여 us-west-2 리전의 my-cluster 클러스터에서 Service 및 Region 키를 가진 태그를 제거합니다.

```
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>
```

## 예약 노드를 통해 비용 관리

하나 이상의 노드를 예약하여 비용을 줄일 수 있습니다. 노드 유형과 예약 기간—(1년 또는 3년)에 따라 예약 노드에 선납금이 부과됩니다. 선결제 요금 외에도 시간당 사용 요금이 부과됩니다. 이 요금은 온디맨드 노드에서 발생하는 시간당 사용 요금보다 훨씬 낮습니다.

예약된 노드가 사용 사례에 대해 비용이 절감되는지 확인하려면 먼저 필요한 노드 수와 노드 크기를 결정합니다. 그런 다음 노드의 사용량을 예측하고 온디맨드 노드와 예약된 노드의 총 비용을 비교합니다. 클러스터에서 예약 노드와 온디맨드 노드를 함께 사용할 수 있습니다. 요금 정보는 [Amazon ElastiCache 요금](#)을 참조하십시오.

AWS Management 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 사용 가능한 예약 노드 제품을 나열하고 구입할 수 있습니다.

예약 노드에 대한 자세한 내용은 [Amazon ElastiCache 예약 캐시 노드](#)를 참조하십시오.

#### 주제

- [표준 예약 노드 캐시 유형](#) (p. 160)
- [이전 예약 노드 캐시 제공](#) (p. 160)
- [예약 노드 상품에 대한 정보 가져오기](#) (p. 161)
- [예약 노드 구입](#) (p. 164)
- [예약 노드에 대한 정보 가져오기](#) (p. 166)

## 표준 예약 노드 캐시 유형

Amazon ElastiCache에서 표준 예약 인스턴스(RI)를 구매할 때는 예약 노드 인스턴스의 기간 동안 특정 캐시 노드 인스턴스 유형 및 AWS 리전에 대해 할인 요금을 이용하는 약정을 구매하는 것입니다. Amazon ElastiCache 예약 노드 인스턴스를 사용하려면 온디맨드 인스턴스와 똑같은 방법으로 새 ElastiCache 노드 인스턴스를 생성해야 합니다.

새 노드 인스턴스는 예약 노드 인스턴스의 사양과 정확히 일치해야 합니다. 새로운 노드 인스턴스의 사양이 계정의 기존 예약 노드 인스턴스와 일치하면 예약 인스턴스에 제공되는 할인 요금이 청구됩니다. 그렇지 않으면 노드 인스턴스에 대해 온디맨드 요금이 청구됩니다. 이 표준 RI는 R5 및 M5 인스턴스 패밀리부터 사용할 수 있습니다.

#### Note

다음에 논의되는 세 가지 제공 유형은 1년과 3년 단위로 사용할 수 있습니다.

#### 제공 유형

**선수금 없음** RI는 선결제 금액 없이 예약된 ElastiCache 인스턴스에 대한 액세스를 제공합니다. 선결제 없음 예약 ElastiCache 인스턴스는 사용되는지 여부와 상관없이 사용 기간 동안 매시간마다 할인된 시간당 요금이 청구됩니다.

**부분 선결제** RI의 경우 예약된 ElastiCache 인스턴스의 일부를 먼저 결제해야 합니다. 결제하지 않은 기간에 대해서는 사용 기간 동안 사용량에 상관없이 할인된 시간당 요금이 청구됩니다. 이 옵션은 다음 단원에서 설명할 이전 Heavy 사용률 옵션을 대신합니다.

**전체 선결제** RI의 경우 RI 사용 기간이 시작될 때 전액 지불해야 합니다. 사용 시간과 관계없이 남은 기간 동안 다른 비용은 발생하지 않습니다.

## 이전 예약 노드 캐시 제공

세 가지의 이전 노드 예약 수준에는 — Heavy 사용률, Medium 사용률 및 Light 사용률이 있습니다. 어느 사용률 수준으로나 1년 또는 3년간 노드를 예약할 수 있습니다. 노드 유형, 사용률 수준 및 예약 기간에 따라 총 비용이 정해집니다. 예약 노드를 구입하기 전에 다양한 모델을 비교하여 예약 노드가 회사에 줄 수 있는 절약 효과를 확인하십시오.

특정 사용률 수준이나 기간으로 구입한 노드를 다른 사용률 수준이나 기간으로 바꿀 수 없습니다.

#### 사용률 수준

Heavy 사용률 예약 노드는 용량 기준이 일관된 워크로드를 가능하게 하거나 되거나 상태가 꾸준한 워크로드를 실행합니다. Heavy 사용률 예약 노드는 높은 선납금 약정이 필요하지만 예약 노드 기간의 79% 이상을 실행하려는 경우 가장 크게 비용을 절감할 수 있습니다(온디맨드 요금의 70%까지). Heavy 사용률 예약 노드의 경우, 일회성 요금을 지불합니다. 이는 노드가 실행 여부에 관계없이 사용 기간 동안 시간당 요금이 더 낮습니다.

Medium 사용률 예약 노드는 많은 기간 예약 노드를 이용할 계획이고 더 저렴한 일회성 요금을 원하거나 노드 사용을 종료할 때 노드 요금 지불을 중지할 수 있기를 원하는 경우 가장 적합한 옵션입니다. Medium 사용률 예약 노드는 예약 노드 기간의 40% 이상을 실행하려는 경우 더욱 비용 효율적인 옵션입니다. 이 옵션은 온디맨드 가격의 최대 64%까지 절감할 수 있습니다. Medium 사용률 예약 노드를 사용하면 Light 사용률 예약 노드를 사용하는 것보다 약간 높은 일회성 요금을 지불하지만 노드를 실행할 때 시간당 사용 요금을 낮출 수 있습니다.

Light 사용률 예약 노드는 매일 두세 시간 또는 일주일에 며칠 정도 실행하는 정기 작업에 적합합니다. Light 사용률 예약 노드를 사용하면 일회성 요금을 지불한 후 노드를 실행할 때 할인된 시간당 요금을 지불하게 됩니다. 노드가 예약된 노드 사용 기간의 17% 이상을 실행 중인 경우, 비용을 절감할 수 있습니다. 예약된 노드의 전체 사용 기간 동안 온디맨드 요금의 최대 56%까지 절감할 수 있습니다.

#### 이전 예약 캐시 노드 상품

제공 유형	선납금	사용료	이점
Heavy 사용률	가장 높음	시간당 가장 낮은 요금. 예약 노드를 사용하고 있는지 상관없이 전체 기간에 적용됩니다.	예상되는 예약 노드 사용률이 3년 약정 기준으로 79% 이상인 경우 전체적인 비용이 가장 낮습니다.
Medium 사용률	Medium	노드를 실행하는 매 시간마다 시간당 사용 요금이 청구됩니다. 노드가 실행되지 않을 때는 시간당 요금이 청구되지 않습니다.	탄력적인 워크로드 또는 3년 약정 기간에 40% 이상의 보통 사용량이 필요한 경우에 적합합니다.
Light 사용률	가장 낮음	노드를 실행하는 매 시간마다 시간당 사용 요금이 청구됩니다. 노드가 실행되지 않을 때는 시간당 요금이 청구되지 않습니다. 모든 제공 유형 중에서 시간당 요금이 가장 높지만 예약 노드가 실행되고 있을 때만 요금이 적용됩니다.	항상 실행할 계획이라면 전체 비용이 가장 높습니다. 그러나, 이는 3년 약정 기간의 약 15% 이상 예약 노드를 자주 사용하지 않는 경우 전체 비용이 가장 낮습니다.
온디맨드 사용(예약 노드 없음)	없음	시간당 요금이 가장 높습니다. 노드가 실행 중일 때마다 적용됩니다.	시간당 비용이 가장 높습니다.

자세한 내용은 [Amazon ElastiCache 요금](#)을 참조하십시오.

## 예약 노드 상품에 대한 정보 가져오기

예약 노드를 구입하기 전에 사용 가능한 예약 노드 상품에 대한 정보를 확인할 수 있습니다.

다음 예제에서는 AWS Management 콘솔, AWS CLI 및 ElastiCache API를 사용하여 사용 가능한 예약 노드 상품의 요금과 정보를 확인하는 방법을 보여줍니다.

#### 주제

- [예약 노드 상품에 대한 정보 가져오기 \(콘솔\) \(p. 162\)](#)
- [예약 노드 상품에 대한 정보 가져오기 \(AWS CLI\) \(p. 162\)](#)

- [예약 노드 상품에 대한 정보 가져오기 \(ElastiCache API\) \(p. 163\)](#)

## 예약 노드 상품에 대한 정보 가져오기 (콘솔)

AWS Management 콘솔을 사용하여 사용 가능한 예약 클러스터 상품의 요금과 그 밖의 정보를 가져오려면 다음 절차를 따르십시오.

사용 가능한 예약 노드 상품에 대한 정보를 가져오려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 Reserved Cache Nodes(예약 캐시 노드)를 선택합니다.
3. [Purchase Reserved Cache Node]를 선택합니다.
4. 제품 설명에서 Memcached를 선택합니다.
5. 사용 가능한 상품을 확인하려면 다음 3개 목록 중에서 선택합니다.
  - 캐시 노드 유형
  - 기간
  - Offering Type(제공 유형)

선택한 후 노드당 비용과 선택 항목의 총 비용이 [Purchase Reserved Cache Nodes] 마법사에 표시됩니다.

6. [Cancel]을 선택하면 이 노드를 구입하지 않으며 요금이 발생하지 않습니다.

## 예약 노드 상품에 대한 정보 가져오기 (AWS CLI)

사용 가능한 예약 노드 상품의 요금과 그 밖의 정보를 가져오려면 명령 프롬프트에 다음 명령을 입력합니다.

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

이 작업은 다음과 비슷한 출력을 생성합니다(JSON 형식).

```
{
 "ReservedCacheNodesOfferings": [
 {
 "OfferingType": "Heavy Utilization",
 "FixedPrice": 4328.0,
 "ReservedCacheNodesOfferingId": "0192caa9-daf2-4159-b1e5-a79bb1916695",
 "UsagePrice": 0.0,
 "RecurringCharges": [
 {
 "RecurringChargeAmount": 0.491,
 "RecurringChargeFrequency": "Hourly"
 }
],
 "ProductDescription": "memcached",
 "Duration": 31536000,
 "CacheNodeType": "cache.r3.4xlarge"
 },
 ***** some output omitted for brevity *****
 {
 "OfferingType": "Heavy Utilization",
 "FixedPrice": 4132.0,
```

```
 "ReservedCacheNodesOfferingId": "fb766e0a-79d7-4e8f-a780-a2a6ed5ed439",
 "UsagePrice": 0.0,
 "RecurringCharges": [
 {
 "RecurringChargeAmount": 0.182,
 "RecurringChargeFrequency": "Hourly"
 }
],
 "ProductDescription": "redis",
 "Duration": 94608000,
 "CacheNodeType": "cache.r3.2xlarge"
 }
]
}
```

자세한 내용은 AWS CLI 참조의 [describe-reserved-cache-nodes-offerings](#)를 참조하십시오.

## 예약 노드 상품에 대한 정보 가져오기 (ElastiCache API)

사용 가능한 예약 노드 상품의 요금과 정보를 가져오려면 `DescribeReservedCacheNodesOfferings` 작업을 호출하십시오.

### 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>
```

이 호출은 다음과 비슷한 출력을 반환합니다.

```
<DescribeReservedCacheNodesOfferingsResponse xmlns="http://elasticache.us-
west-2.amazonaws.com/doc/2013-06-15/">
 <DescribeReservedCacheNodesOfferingsResult>
 <ReservedCacheNodesOfferings>
 <ReservedCacheNodesOffering>
 <Duration>31536000</Duration>
 <OfferingType>Medium Utilization</OfferingType>
 <CurrencyCode>USD</CurrencyCode>
 <RecurringCharges/>
 <FixedPrice>1820.0</FixedPrice>
 <ProductDescription>memcached</ProductDescription>
 <UsagePrice>0.368</UsagePrice>
 <ReservedCacheNodesOfferingId>438012d3-4052-4cc7-b2e3-8d3372e0e706</
ReservedCacheNodesOfferingId>
 <CacheNodeType>cache.m1.large</CacheNodeType>
 </ReservedCacheNodesOffering>
 </ReservedCacheNodesOffering>

 (...some output omitted for brevity...)

 </ReservedCacheNodesOffering>
</DescribeReservedCacheNodesOfferingsResult>
</ResponseMetadata>
```

```
<RequestId>5e4ec40b-2978-11e1-9e6d-771388d6ed6b</RequestId>
</ResponseMetadata>
</DescribeReservedCacheNodesOfferingsResponse>
```

자세한 내용은 ElastiCache API의 [DescribeReservedCacheNodesOfferings](#)를 참조하십시오.

## 예약 노드 구입

다음 예제에서는 AWS Management 콘솔, AWS CLI 및 ElastiCache API를 사용하여 예약 노드 상품을 구입하는 방법을 설명합니다.

### Important

이 단원의 예제에 따라 예약할 수 없는 AWS 계정에 요금이 부과됩니다.

### 주제

- [예약 노드 구입 \(콘솔\) \(p. 164\)](#)
- [예약 노드 구입 \(AWS CLI\) \(p. 165\)](#)
- [예약 노드 구입 \(ElastiCache API\) \(p. 165\)](#)

## 예약 노드 구입 (콘솔)

이 예제에서는 예약 노드 ID가 myreservationID인 특정 예약 노드 상품 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f를 구입하는 방법을 보여줍니다.

다음 절차에서는 AWS Management 콘솔을 사용하여 상품 ID로 예약 노드 상품을 구입합니다.

### 예약 노드를 구입하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 목록에서 [Reserved Cache Nodes] 링크를 선택합니다.
3. [Purchase Reserved Cache Node] 버튼을 선택합니다.
4. [Product Description] 드롭다운 목록 상자에서 노드 유형을 선택합니다.
5. [Cache Node Class] 드롭다운 목록 상자에서 노드 클래스를 선택합니다.
6. [Term] 드롭다운 목록 상자에서 노드를 예약할 기간을 선택합니다.
7. 다음 중 하나를 수행하십시오.
  - [Offering Type] 드롭다운 목록 상자에서 제공 유형을 선택합니다.
  - [Reserved Cache Node ID] 텍스트 상자에 예약 노드 ID를 입력합니다.

### Note

예약 캐시 노드 ID는 고객마다 고유한 식별자이며 이 예약을 추적하는 데 사용됩니다. 이 상자를 비워 두면 ElastiCache에서 자동으로 예약 식별자를 생성합니다.

8. [Next] 버튼을 선택합니다.

[Purchase Reserved Cache Node] 대화 상자에 선택한 예약 노드 속성에 대한 요약 내용과 지불 기한이 표시됩니다.

9. 계속하려면 [Yes, Purchase] 버튼을 선택하고 예약 노드를 구입합니다.

### Important

[Yes, Purchase]를 선택하면 선택한 예약 노드에 요금이 부과됩니다. 요금이 부과되지 않도록 하려면 [Cancel]을 선택합니다.

## 예약 노드 구입 (AWS CLI)

다음 예제에서는 예약 노드 ID가 myreservationID인 특정 예약 클러스터 상품 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f를 구입하는 방법을 보여줍니다.

명령 프롬프트에서 다음 명령을 입력합니다.

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

이 명령은 다음과 비슷한 출력을 반환합니다.

RESERVATION	ReservationId	Class	Start Time	Duration	Fixed
Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	myreservationid	cache.m1.small	2013-12-19T00:30:23.247Z	1y	455.00
USD	0.092 USD	1	payment-pending	memcached	Medium Utilization

자세한 내용은 AWS CLI 참조의 [purchase-reserved-cache-nodes-offering](#)을 참조하십시오.

## 예약 노드 구입 (ElastiCache API)

다음 예제에서는 예약 클러스터 노드 ID가 myreservationID인 특정 예약 상품 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f를 구입하는 방법을 보여줍니다.

다음 파라미터와 함께 PurchaseReservedCacheNodesOffering 작업을 호출합니다.

- 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=AWS4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

이 호출은 다음과 비슷한 출력을 반환합니다.

```
<PurchaseReservedCacheNodesOfferingResponse xmlns="http://elasticache.us-
west-2.amazonaws.com/doc/2013-06-15/">
 <PurchaseReservedCacheNodesOfferingResult>
 <ReservedCacheNode>
 <OfferingType>Medium Utilization</OfferingType>
 <CurrencyCode>USD</CurrencyCode>
 <RecurringCharges/>
 <ProductDescription>memcached</ProductDescription>
 <ReservedCacheNodesOfferingId>649fd0c8-cf6d-47a0-bfa6-060f8e75e95f</
ReservedCacheNodesOfferingId>
 <State>payment-pending</State>
 <ReservedCacheNodeId>myreservationID</ReservedCacheNodeId>
 <CacheNodeCount>10</CacheNodeCount>
 <StartTime>2013-07-18T23:24:56.577Z</StartTime>
 <Duration>31536000</Duration>
 <FixedPrice>123.0</FixedPrice>
 <UsagePrice>0.123</UsagePrice>
 <CacheNodeType>cache.m1.small</CacheNodeType>
 </ReservedCacheNode>
 </PurchaseReservedCacheNodesOfferingResult>
 <ResponseMetadata>
 <RequestId>7f099901-29cf-11e1-bd06-6fe008f046c3</RequestId>
 </ResponseMetadata>
</PurchaseReservedCacheNodesOfferingResponse>
```

자세한 내용은 ElastiCache API 참조의 [PurchaseReservedCacheNodesOffering](#)을 참조하십시오.

## 예약 노드에 대한 정보 가져오기

AWS Management 콘솔, AWS CLI 및 ElastiCache API를 사용하여 구입한 예약 노드에 대한 정보를 가져올 수 있습니다.

주제

- [예약 노드에 대한 정보 가져오기 \(콘솔\)](#) (p. 166)
- [예약 노드에 대한 정보 가져오기 \(AWS CLI\)](#) (p. 166)
- [예약 노드에 대한 정보 가져오기 \(ElastiCache API\)](#) (p. 167)

## 예약 노드에 대한 정보 가져오기 (콘솔)

다음 절차에서는 AWS Management 콘솔을 사용하여 구입한 예약 노드에 대한 정보를 가져오는 방법을 설명합니다.

구입한 예약 노드에 대한 정보를 가져오려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 목록에서 [Reserved Cache Nodes] 링크를 선택합니다.

계정의 예약 노드가 [Reserved Cache Nodes] 목록에 나타난다. 목록에서 예약 노드를 선택하면 콘솔 아래쪽의 세부 정보 창에 예약 노드에 대한 자세한 정보가 표시됩니다.

## 예약 노드에 대한 정보 가져오기 (AWS CLI)

AWS 계정의 예약 노드에 대한 정보를 가져오려면 명령 프롬프트에 명령을 입력하십시오.

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

이 작업은 다음과 비슷한 출력을 생성합니다(JSON 형식).

```
{
 "ReservedCacheNodeId": "myreservationid",
 "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
 "CacheNodeType": "cache.m1.small",
 "Duration": "31536000",
 "ProductDescription": "memcached",
 "OfferingType": "Medium Utilization",
 "MaxRecords": 0
}
```

자세한 내용은 AWS CLI 참조의 [describe-reserved-cache-nodes](#)를 참조하십시오.

## 예약 노드에 대한 정보 가져오기 (ElastiCache API)

AWS 계정의 예약 노드에 대한 정보를 가져오려면 DescribeReservedCacheNodes 작업을 호출하십시오.

### Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeReservedCacheNodes
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

이 호출은 다음과 비슷한 출력을 반환합니다.

```
<DescribeReservedCacheNodesResponse xmlns="http://elasticache.us-west-2.amazonaws.com/doc/2013-06-15/">
 <DescribeReservedCacheNodesResult>
 <ReservedCacheNodes>
 <ReservedCacheNode>
 <OfferingType>Medium Utilization</OfferingType>
 <CurrencyCode>USD</CurrencyCode>
 <RecurringCharges/>
 <ProductDescription>memcached</ProductDescription>
 <ReservedCacheNodesOfferingId>649fd0c8-cf6d-47a0-bfa6-060f8e75e95f</ReservedCacheNodesOfferingId>
 <State>payment-failed</State>
 <ReservedCacheNodeId>myreservationid</ReservedCacheNodeId>
 <CacheNodeCount>1</CacheNodeCount>
 <StartTime>2010-12-15T00:25:14.131Z</StartTime>
 <Duration>31536000</Duration>
 <FixedPrice>227.5</FixedPrice>
 <UsagePrice>0.046</UsagePrice>
 <CacheNodeType>cache.m1.small</CacheNodeType>
 </ReservedCacheNode>
 </ReservedCacheNode>
 </DescribeReservedCacheNodesResult>
</DescribeReservedCacheNodesResponse>
```

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
예약 노드에 대한 정보 가져오기

---

```
(...some output omitted for brevity...)

</ReservedCacheNode>
</ReservedCacheNodes>
</DescribeReservedCacheNodesResult>
<ResponseMetadata>
 <RequestId>23400d50-2978-11e1-9e6d-771388d6ed6b</RequestId>
</ResponseMetadata>
</DescribeReservedCacheNodesResponse>
```

자세한 내용은 ElastiCache API 참조의 [DescribeReservedCacheNodes](#)를 참조하십시오.

# Amazon ElastiCache의 보안

AWS에서는 클라우드 보안을 가장 중요하게 생각합니다. AWS 고객은 보안에 매우 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 귀하의 공동 책임입니다. **책임 분담 모델**은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드의 보안 – AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호합니다. AWS는 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사원은 정기적으로 **AWS 규제 준수 프로그램**의 일환으로 보안 효과를 테스트하고 검증합니다. Amazon ElastiCache에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 **AWS 규정 준수 프로그램 제공 범위 내 서비스**를 참조하십시오.
- 클라우드 내 보안 – 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon ElastiCache 사용 시 책임 분담 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 충족하도록 Amazon ElastiCache를 구성하는 방법을 보여줍니다. 또한 Amazon ElastiCache 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 배우게 됩니다.

## 주제

- [Amazon ElastiCache의 데이터 보호 \(p. 169\)](#)
- [인터넷워크 트래픽 개인 정보 \(p. 170\)](#)
- [Amazon ElastiCache의 Identity and Access Management\(IAM\) \(p. 194\)](#)
- [Amazon ElastiCache 규정 준수 확인 \(p. 212\)](#)
- [Amazon ElastiCache의 복원성 \(p. 212\)](#)
- [AWS ElastiCache의 인프라 보안 \(p. 214\)](#)

## Amazon ElastiCache의 데이터 보호

Amazon ElastiCache는 AWS **공동 책임 모델**을 준수하며, 여기에는 데이터 보호 관련 규정 및 지침이 포함됩니다. AWS는 모든 AWS 서비스를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. AWS는 고객 콘텐츠 및 개인 데이터의 처리를 위한 보안 구성 제어 등 이 인프라에서 호스팅되는 데이터에 대한 제어권을 유지합니다. 데이터 컨트롤러 또는 데이터 처리자의 역할을 담당하는 AWS 고객과 APN 파트너는 AWS Cloud에 올린 모든 개인 데이터에 대해 책임을 집니다.

데이터를 보호하려면 AWS 계정 자격 증명을 보호하고 AWS Identity and Access Management(IAM)을 사용해 개별 사용자 계정을 설정하는 것이 좋습니다. 이러한 접근 방식에서는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정마다 멀티 팩터 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 개인 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.

이름 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마십시오. 여기에는 Amazon ElastiCache 또는 기타 AWS 서비스에서 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 작업하는 경우가 포함됩니다. Amazon ElastiCache 또는 기타 서비스에 입력하는 모든 데이터는 진단 로그에 포함하기 위해 선택될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 증명 정보를 URL에 포함시키지 마십시오.

데이터 보호에 관한 자세한 내용은 AWS 보안 블로그에서 [AWS 공동 책임 모델 및 GDPR](#) 블로그 게시물을 참조하십시오.

## 인터넷워크 트래픽 개인 정보

Amazon ElastiCache는 다음 기술을 사용하여 캐시 데이터 보안을 유지하고 무단 액세스로부터 보호합니다.

- [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#)은 설치에 필요한 보안 그룹 유형을 설명합니다.
- 사용자, 그룹 및 역할의 작업을 부여하고 제한하기 위한 [Amazon ElastiCache의 Identity and Access Management\(IAM\) \(p. 194\)](#)

## Amazon VPC 및 ElastiCache 보안

데이터 보안이 중요하기 때문에 ElastiCache는 데이터를 액세스할 수 있는 대상을 제어하는 수단을 제공합니다. 데이터에 대한 액세스를 제어하는 방법은 클러스터를 Amazon Virtual Private Cloud(Amazon VPC)에서 시작했는지 또는 Amazon EC2-Classic에서 시작했는지에 따라 다릅니다.

### Important

ElastiCache 클러스터 시작을 위해 Amazon EC2-Classic을 사용하던 방식은 사용이 중지되었습니다. 모든 현재 세대 노드는 Amazon Virtual Private Cloud에서만 시작됩니다.

### 주제

- [ElastiCache 및 Amazon VPC 이해 \(p. 171\)](#)
- [Amazon VPC에 있는 ElastiCache 클러스터에 액세스하기 위한 액세스 패턴 \(p. 174\)](#)
- [Virtual Private Cloud\(VPC\) 생성 \(p. 180\)](#)
- [캐시 서브넷 그룹 생성 \(p. 182\)](#)
- [Amazon VPC에서 캐시 클러스터 생성 \(p. 182\)](#)
- [Amazon VPC에서 실행되는 클러스터에 연결 \(p. 182\)](#)

Amazon Virtual Private Cloud(Amazon VPC) 서비스는 전통적인 데이터 센터와 매우 유사한 가상 네트워크를 정의합니다. Amazon VPC를 구성할 때 그 IP 주소 범위를 선택하고 서브넷을 생성하고 라우팅 테이블, 네트워크 게이트웨이 및 보안 설정을 구성할 수 있습니다. 또한 Amazon VPC 보안 그룹을 사용하여 가상 네트워크에 캐시 클러스터를 추가하고 캐시 클러스터에 대한 액세스 권한을 제어할 수 있습니다.

이 단원에서는 Amazon VPC의 ElastiCache 클러스터를 수동으로 구성하는 방법을 설명합니다. 이 정보는 ElastiCache 및 Amazon VPC가 연동되는 방식을 더 깊이 이해하고자 하는 사용자를 대상으로 합니다.

### 주제

- [ElastiCache 및 Amazon VPC 이해 \(p. 171\)](#)
- [Amazon VPC에 있는 ElastiCache 클러스터에 액세스하기 위한 액세스 패턴 \(p. 174\)](#)
- [Virtual Private Cloud\(VPC\) 생성 \(p. 180\)](#)
- [캐시 서브넷 그룹 생성 \(p. 182\)](#)
- [Amazon VPC에서 캐시 클러스터 생성 \(p. 182\)](#)
- [Amazon VPC에서 실행되는 클러스터에 연결 \(p. 182\)](#)

## ElastiCache 및 Amazon VPC 이해

ElastiCache은 Amazon Virtual Private Cloud (Amazon VPC)과 완벽하게 통합됩니다. ElastiCache 사용자의 경우 이는 다음을 의미합니다.

- AWS 계정에서 EC2-VPC 플랫폼만 지원하는 경우 ElastiCache는 항상 Amazon VPC에서 클러스터를 시작합니다.
- AWS를 처음 이용하는 경우 클러스터가 Amazon VPC에 배포됩니다. 기본 VPC가 자동으로 생성됩니다.
- 기본 VPC가 있고 클러스터를 시작할 때 서브넷을 지정하지 않으면 클러스터가 기본 Amazon VPC에서 시작합니다.

자세한 정보는 [Detecting Your Supported Platforms and Whether You Have a Default VPC](#)를 참조하십시오.

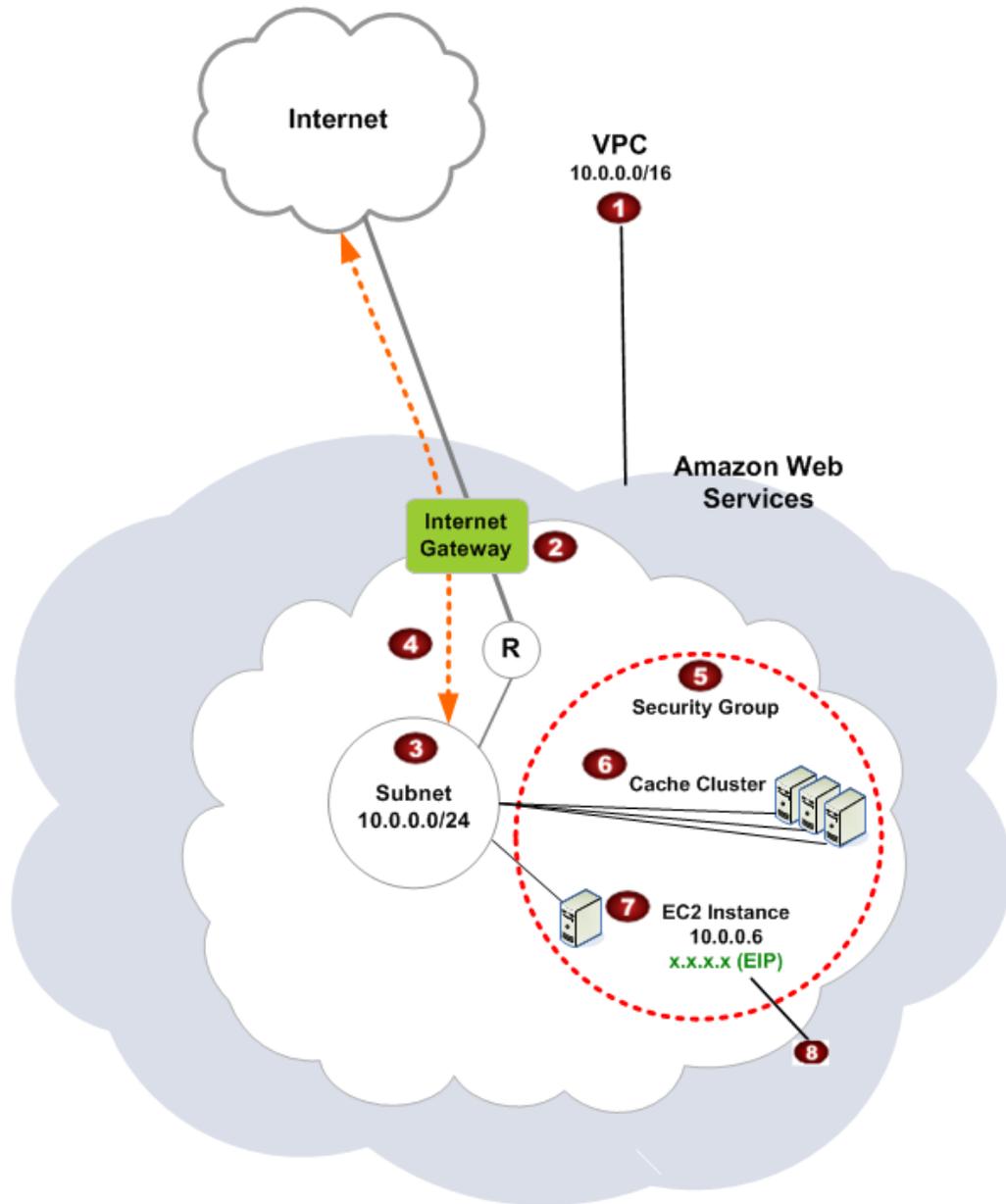
Amazon Virtual Private Cloud를 사용하면 기존의 데이터 센터와 매우 유사한 AWS 클라우드에 가상 네트워크를 생성할 수 있습니다. IP 주소 범위 선택, 서브넷 생성 및 라우팅 테이블, 네트워크 게이트웨이, 보안 설정 구성을 포함하여 Amazon VPC를 구성할 수 있습니다.

ElastiCache의 기본 기능은 가상 프라이빗 클라우드에서 동일합니다. ElastiCache는 클러스터가 Amazon VPC의 내부 또는 외부에 배포되는 여부와 관계없이 소프트웨어 업그레이드, 패치 적용, 장애 탐지 및 복구를 관리합니다.

Amazon VPC 외부에 배포된 ElastiCache 캐시 노드에는 엔드포인트/DNS 이름이 확인되는 IP 주소가 할당됩니다. 이는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 연결을 제공합니다. ElastiCache 클러스터를 Amazon VPC 프라이빗 서브넷으로 실행할 때 모든 캐시 노드는 그 서브넷 내의 프라이빗 IP 주소에 할당됩니다.

### Amazon VPC에서의 ElastiCache 개요

다음 다이어그램과 표에는 Amazon VPC에서 시작되는 ElastiCache 클러스터 및 Amazon EC2 인스턴스와 함께 Amazon VPC 환경에 대한 설명이 나와 있습니다.



<b>1</b>	Amazon VPC는 자체 IP 주소 블록이 할당된 AWS 클라우드의 격리된 부분입니다.
<b>2</b>	인터넷 게이트웨이는 Amazon VPC를 인터넷에 직접 연결하고 Amazon VPC 외부에서 실행되는 Amazon Simple Storage Service(Amazon S3)와 같은 다른 AWS 리소스에 대한 액세스를 제공합니다.
<b>3</b>	Amazon VPC 서브넷은 보안 및 운영상의 필요에 따라 AWS 리소스를 격리할 수 있는 Amazon VPC의 IP 주소 범위 세그먼트입니다.
<b>4</b>	Amazon VPC의 라우팅 테이블은 서브넷과 인터넷 간 네트워크 트래픽을 지시합니다. Amazon VPC에는 이 다이어그램에서 원 안에 R로 표시된 라우터가 내재되어 있습니다.

<b>5</b>	Amazon VPC 보안 그룹은 ElastiCache 클러스터와 Amazon EC2 인스턴스의 인바운드 및 아웃바운드 트래픽을 제어합니다.
<b>6</b>	서브넷에서 ElastiCache 클러스터를 실행할 수 있습니다. 캐시 노드는 서브넷 주소 범위의 프라이빗 IP 주소를 가집니다.
<b>7</b>	또한 서브넷에서 Amazon EC2 인스턴스를 시작할 수 있습니다. 각각의 Amazon EC2 인스턴스는 서브넷 주소 범위의 프라이빗 IP 주소를 가집니다. Amazon EC2 인스턴스를 동일한 서브넷의 모든 캐시 노드에 연결할 수 있습니다.
<b>8</b>	Amazon VPC의 Amazon EC2 인스턴스를 인터넷에 연결하려면 인스턴스에 탄력적 IP 주소라는 고정 퍼블릭 주소를 할당해야 합니다.

## ElastiCache 배포에서 EC2 Classic 대신 Amazon VPC를 사용해야 하는 이유

인스턴스를 Amazon VPC로 시작하면 다음을 수행할 수 있습니다.

- 인스턴스의 시작/중지에 상관 없이 유지되는 고정 IP 주소 할당.
- 인스턴스에 여러 개의 IP 주소 할당이 가능합니다.
- 네트워크 인터페이스를 정의하고, 하나 혹은 그 이상의 네트워크 인터페이스를 귀하의 인스턴스에 설치 가능합니다.
- 인스턴스를 실행하는 동안 인스턴스의 보안 그룹 멤버십을 변경할 수 있습니다.
- 인스턴스의 인바운드 트래픽 제어(인그레스 필터링)뿐만 아니라 아웃바운드 트래픽도 제어(인그레스 필터링)할 수 있습니다.
- 네트워크 액세스 제어 리스트(ACL)를 통해, 인스턴스에 대해 액세스 제어가 한 단계 더 강화되었습니다.
- 단일 테넌트 하드웨어에서 인스턴스 실행.

Amazon EC2 Classic, 기본 VPC 및 기본 외 VPC를 비교하려면 [VPC-Classical과 EC2-VPC의 차이점](#)을 참조하십시오.

Amazon VPC는 비전용 Amazon EC2 인스턴스를 허용해야 합니다. 전용 인스턴스 테넌시로 구성된 Amazon VPC에서는 ElastiCache를 사용할 수 없습니다.

## 사전 조건

Amazon VPC에 ElastiCache 클러스터를 생성하려면 Amazon VPC가 다음 요구 사항을 충족해야 합니다.

- Amazon VPC는 비전용 Amazon EC2 인스턴스를 허용해야 합니다. 전용 인스턴스 테넌시로 구성된 Amazon VPC에서는 ElastiCache를 사용할 수 없습니다.
- Amazon VPC에 대해 캐시 서브넷 그룹을 정의해야 합니다. ElastiCache는 캐시 서브넷 그룹을 사용하여 캐시 노드에 연결된 서브넷 내의 서브넷 및 IP 주소를 선택할 수 있습니다.
- Amazon VPC에 대해 캐시 보안 그룹을 정의하거나 제공된 기본값을 사용할 수 있습니다.
- 각 서브넷의 CIDR 블록은 유지 관리 작업에서 ElastiCache에 사용할 여분의 IP 주소를 제공할 수 있을 만큼 충분히 커야 합니다.

## 라우팅 및 보안

Amazon VPC에서 라우팅을 구성하여 트래픽 흐름(예: 인터넷 게이트웨이, 가상 프라이빗 게이트웨이)을 제어할 수 있습니다. 인터넷 게이트웨이를 통해 Amazon VPC를 Amazon VPC에서 실행되지 않는 다른 AWS 리소스에 직접 액세스할 수 있습니다. 조직의 로컬 네트워크에 연결된 가상 사설 게이트웨이만을 사용하도록 선택한 경우, VPN을 통해 인터넷 바운드 트래픽을 라우팅하고 출구를 제어하기 위한 로컬 보안 정책 및 방화

벽을 사용할 수 있습니다. 이 경우 인터넷을 통해 AWS 리소스에 액세스할 때 대역폭 요금이 추가로 부과됩니다.

Amazon VPC 보안 그룹을 사용하여 Amazon VPC에서 ElastiCache 클러스터 및 Amazon EC2 인스턴스를 보호할 수 있습니다. 보안 그룹은 서브넷 레벨이 아닌 인스턴스 레벨에서 방화벽처럼 작동합니다.

#### Note

캐시 노드를 재부팅하는 경우 기본 IP 주소가 변경될 수 있으므로 캐시 노드에 연결할 때 DNS 이름을 사용하는 것이 좋습니다.

## Amazon VPC 문서

Amazon VPC에는 Amazon VPC를 생성하고 사용하는 방법을 설명하는 자체 문서 세트가 있습니다. 다음 테이블은 Amazon VPC 지침의 링크를 제공합니다.

설명	설명서
Amazon VPC 사용을 시작하는 방법	<a href="#">Amazon VPC 시작 안내서</a>
AWS Management 콘솔을 통해 Amazon VPC를 사용하는 방법	<a href="#">Amazon VPC 사용 설명서</a>
모든 Amazon VPC 명령의 전체 설명	<a href="#">Amazon EC2 명령줄 레퍼런스</a> (Amazon VPC 명령은 Amazon EC2 참조의 일부임)
Amazon VPC API 작업, 데이터 형식 및 오류의 전체 설명	<a href="#">Amazon EC2 API Reference</a> (Amazon VPC API 작업은 Amazon EC2 참조의 일부임)
선택적인 IPsec VPN 연결 사용자 측의 게이트웨이를 구성하는 데 필요한 네트워크 관리자를 위한 정보	<a href="#">AWS Site-to-Site VPN 네트워크 관리자 안내서</a>

Amazon Virtual Private Cloud에 대한 자세한 내용은 [Amazon Virtual Private Cloud](#) 단원을 참조하십시오.

## Amazon VPC에 있는 ElastiCache 클러스터에 액세스하기 위한 액세스 패턴

Amazon ElastiCache에서는 Amazon VPC에 있는 클러스터에 액세스할 수 있도록 다음 시나리오를 지원합니다.

#### 목차

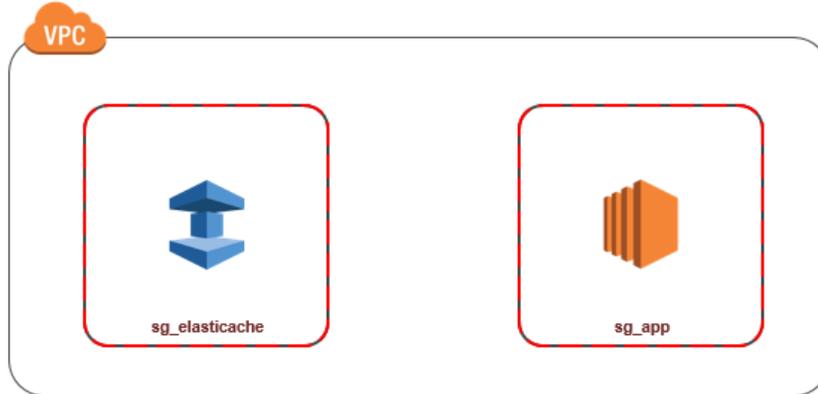
- [ElastiCache 클러스터와 Amazon EC2 인스턴스가 같은 Amazon VPC에 있는 경우 클러스터 액세스 \(p. 175\)](#)
- [ElastiCache 클러스터와 Amazon EC2 인스턴스가 다른 Amazon VPC에 있는 경우 클러스터 액세스 \(p. 176\)](#)
  - [ElastiCache 클러스터와 Amazon EC2 인스턴스가 같은 리전의 다른 Amazon VPC에 있는 경우 클러스터 액세스 \(p. 176\)](#)
  - [ElastiCache 클러스터와 Amazon EC2 인스턴스가 다른 리전의 다른 Amazon VPC에 있는 경우 클러스터 액세스 \(p. 177\)](#)
- [고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스 \(p. 178\)](#)
  - [VPN 연결을 사용하여 고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스 \(p. 178\)](#)

- [Direct Connect를 사용하여 고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스 \(p. 179\)](#)

## ElastiCache 클러스터와 Amazon EC2 인스턴스가 같은 Amazon VPC에 있는 경우 클러스터 액세스

가장 일반적인 사용 사례는 EC2 인스턴스에 배포된 애플리케이션이 같은 VPC에 있는 클러스터에 연결해야 하는 경우입니다.

다음은 이 시나리오를 설명한 다이어그램입니다



동일한 VPC에서 EC2 인스턴스와 DB 인스턴스 간 액세스를 관리하는 가장 간단한 방법은 다음과 같습니다.

1. 클러스터의 VPC 보안 그룹을 만듭니다. 이 보안 그룹을 사용해 클러스터 인스턴스에 대한 액세스를 제한할 수 있습니다. 예를 들어, 클러스터를 만들 때 할당한 포트와 클러스터에 액세스할 때 이용할 IP 주소를 사용해 TCP 액세스를 허용하는 이 보안 그룹의 사용자 지정 규칙을 만들 수 있습니다.

Memcached 클러스터의 기본 포트는 11211입니다.

2. EC2 인스턴스(웹 및 애플리케이션 서버)의 VPC 보안 그룹을 만듭니다. 이 보안 그룹은 필요할 경우 VPC의 라우팅 테이블을 통한 EC2 인스턴스 액세스를 허용할 수 있습니다. 예를 들어, 이 보안 그룹에서 TCP가 포트 22를 통해 EC2 인스턴스에 액세스하도록 허용하는 규칙을 설정할 수 있습니다.
3. 클러스터에 대한 보안 그룹에서 EC2 인스턴스에 대해 생성한 보안 그룹으로부터의 연결을 허용하는 사용자 지정 규칙을 만듭니다. 그러면 보안 그룹의 모든 구성원이 DB 인스턴스에 액세스하도록 허용됩니다.

VPC 보안 그룹에서 다른 보안 그룹으로부터의 연결을 허용하는 규칙을 만들려면

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/vpc>에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. 클러스터 인스턴스에 사용할 보안 그룹을 선택하거나 만듭니다. 규칙 추가를 선택합니다. 이 보안 그룹은 다른 보안 그룹 멤버에 대한 액세스를 허용합니다.
4. [Type]에서 [Custom TCP Rule]을 선택합니다.
  - a. [Port Range]에 대해 클러스터를 만들 때 사용한 포트를 지정합니다.

Memcached 클러스터의 기본 포트는 11211입니다.
  - b. [Source] 상자에 보안 그룹 ID를 입력합니다. 목록에서 Amazon EC2 인스턴스에 사용할 보안 그룹을 선택합니다.
5. 완료되면 [Save]를 선택합니다.

## Edit inbound rules

Type <small>i</small>	Protocol <small>i</small>	Port Range <small>i</small>	Source <small>i</small>
Custom TCP Rule	TCP	6379	Custom sg_app

[Add Rule](#)

### ElastiCache 클러스터와 Amazon EC2 인스턴스가 다른 Amazon VPC에 있는 경우 클러스터 액세스

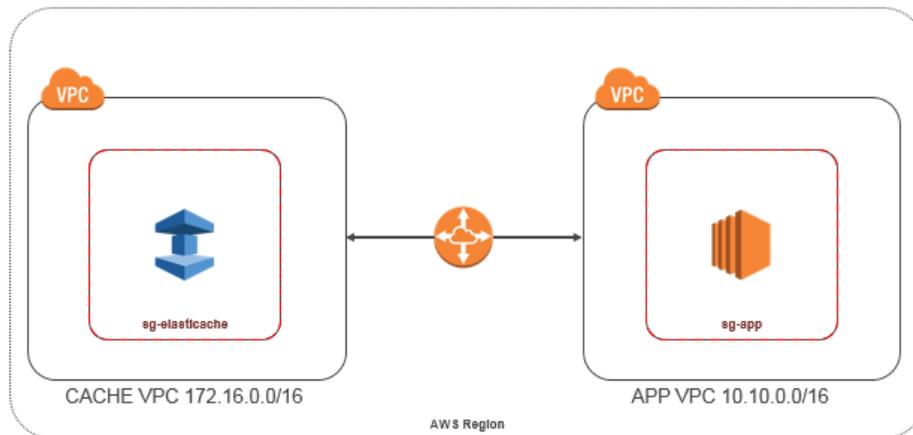
클러스터가 액세스에 사용할 EC2 인스턴스와 다른 VPC에 있는 경우 여러 가지 방법으로 DB 인스턴스에 액세스할 수 있습니다. 클러스터와 EC2 인스턴스가 서로 다른 VPC에 있지만 리전은 동일한 경우 VPC 피어링을 사용할 수 있습니다. 클러스터와 EC2 인스턴스가 서로 다른 리전에 있으면 리전 간에 VPN 연결을 만들 수 있습니다.

#### 주제

- [ElastiCache 클러스터와 Amazon EC2 인스턴스가 같은 리전의 다른 Amazon VPC에 있는 경우 클러스터 액세스 \(p. 176\)](#)
- [ElastiCache 클러스터와 Amazon EC2 인스턴스가 다른 리전의 다른 Amazon VPC에 있는 경우 클러스터 액세스 \(p. 177\)](#)

### ElastiCache 클러스터와 Amazon EC2 인스턴스가 같은 리전의 다른 Amazon VPC에 있는 경우 클러스터 액세스

다음 다이어그램에서는 Amazon VPC 피어링 연결을 사용하여 같은 리전의 서로 다른 Amazon VPC에 있는 Amazon EC2 인스턴스에서 클러스터에 액세스하는 과정을 보여줍니다.



#### 같은 리전의 다른 Amazon VPC에 있는 Amazon EC2 인스턴스가 클러스터 액세스 - VPC 피어링 연결

VPC 피어링 연결은 프라이빗 IP 주소를 사용하여 두 VPC 간에 트래픽을 라우팅할 수 있도록 하기 위한 두 VPC 사이의 네트워킹 연결입니다. 동일한 네트워크에 속하는 경우와 같이 VPC의 인스턴스가 서로 통신

할 수 있습니다. 자체 Amazon VPC간의 VPC 피어링 연결, 또는 단일 리전 내에 있는 다른 AWS 계정에서 Amazon VPC와의 VPC 피어링 연결을 만들 수 있습니다. Amazon VPC 피어링에 대한 자세한 내용은 [VPC 설명서](#)를 참조하십시오.

피어링으로 다른 Amazon VPC에 있는 클러스터에 액세스하려면

1. 두 VPC에 겹치는 IP 범위가 없거나 이 VPC를 피어링할 수 없어야 합니다.
2. 두 VPC를 피어링합니다. 자세한 내용은 [Amazon VPC 피어링 연결 생성 및 수락](#)을 참조하십시오.
3. 라우팅 테이블을 업데이트합니다. 자세한 내용은 [VPC 피어링 연결을 위한 라우팅 테이블 업데이트](#)를 참조하십시오.

앞에 나온 다이어그램의 예제에 대한 라우팅 테이블은 다음과 같습니다. pcx-a894f1c1이 피어링 연결입니다.

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

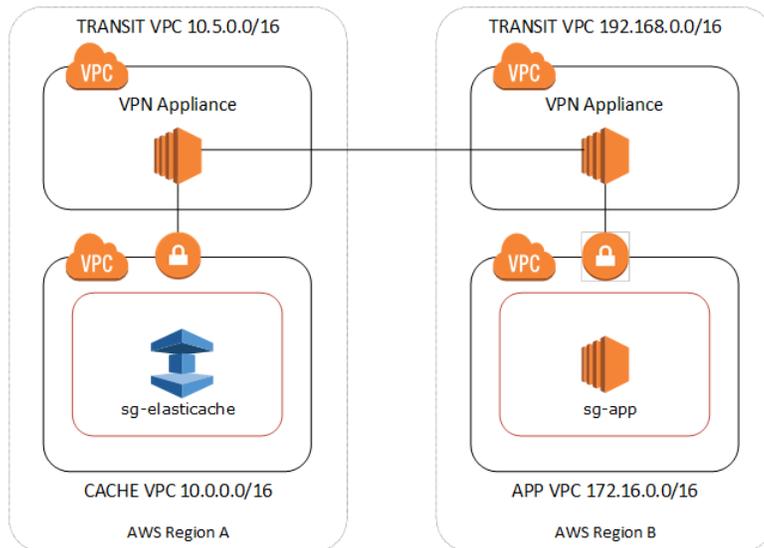
VPC 라우팅 테이블

4. ElastiCache 클러스터의 보안 그룹을 수정하여 피어링된 VPC의 애플리케이션 보안 그룹에서 들어오는 인바운드 연결을 허용합니다. 자세한 내용은 [피어 VPC 보안 그룹 참조](#)를 참조하십시오.

피어링 연결을 통해 클러스터에 액세스하면 데이터 전송 비용이 추가로 발생합니다.

### ElastiCache 클러스터와 Amazon EC2 인스턴스가 다른 리전의 다른 Amazon VPC에 있는 경우 클러스터 액세스

지리적으로 떨어져 있는 여러 VPC와 원격 네트워크를 연결하는 일반적인 전략 하나는 글로벌 네트워크 전송 센터 역할을 하는 전송 VPC를 만드는 것입니다. 전송 VPC는 네트워크 관리를 간소화하고 여러 VPC와 원격 네트워크를 연결하는 데 필요한 연결 수를 최소화합니다. 이 디자인은 기존의 방식으로 공동 배치 전송 허브에 실제 존재를 만들거나 물리적 네트워크 장비를 배포하지 않고 가상으로 구현되므로 시간, 노력, 비용을 아낄 수 있습니다.



다른 리전의 다른 VPC를 지나는 연결

전송 Amazon VPC가 설정되면 한 리전의 "스포크" VPC에 배포된 애플리케이션이 다른 리전의 "스포크" VPC에 있는 ElastiCache 클러스터에 연결할 수 있습니다.

다른 리전의 다른 VPC에 있는 클러스터에 액세스하려면

1. 전송 VPC 솔루션을 배포합니다. 자세한 내용은 [How do I build a global transit network on AWS?](#) 단원을 참조하십시오.
2. 앱 및 캐시 VPC에서 VPC 라우팅 테이블을 업데이트하여 VGW(가상 프라이빗 게이트웨이) 및 VPN 어플라이언스를 통해 트래픽을 라우팅합니다. BGP(Border Gateway Protocol)를 사용하는 동적 라우팅의 경우 라우팅이 자동으로 전파됩니다.
3. ElastiCache 클러스터의 보안 그룹을 수정하여 애플리케이션 인스턴스 IP 범위에서 들어오는 인바운드 연결을 허용합니다. 이 시나리오에는 애플리케이션 서버 보안 그룹을 참조할 수 없습니다.

여러 리전의 클러스터에 액세스하면 네트워크 지연 시간이 생기고 교차 리전 데이터 전송 비용이 추가로 발생합니다.

## 고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스

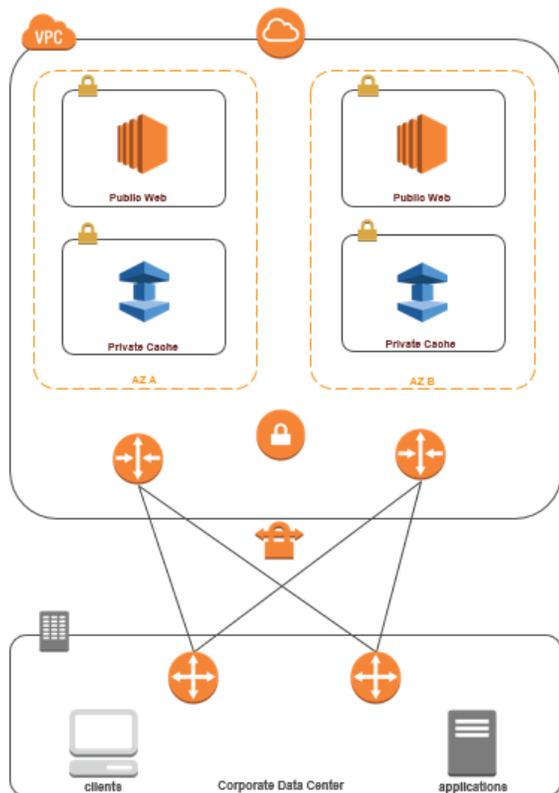
고객 데이터 센터의 클라이언트나 애플리케이션이 VPC의 ElastiCache 클러스터에 액세스해야 하는 하이브리드 아키텍처도 가능한 시나리오입니다. VPN이나 Direct Connect를 통해 고객의 VPC와 데이터 센터 간에 연결을 제공하여 이 시나리오도 지원됩니다.

주제

- [VPN 연결을 사용하여 고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스 \(p. 178\)](#)
- [Direct Connect를 사용하여 고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스 \(p. 179\)](#)

## VPN 연결을 사용하여 고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스

다음 다이어그램에서는 VPN 연결을 사용하여 회사 네트워크에서 실행되는 애플리케이션에서 ElastiCache 클러스터에 액세스하는 과정을 보여줍니다.



VPN를 통해 데이터 센터에서 ElastiCache에 연결

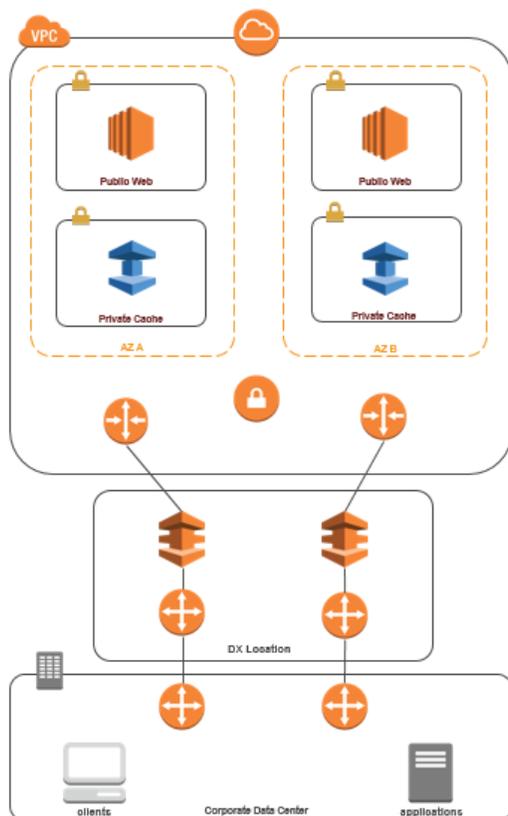
VPN 연결을 통해 온-프레미스 애플리케이션에서 VPC의 클러스터에 액세스하려면

1. 하드웨어 가상 프라이빗 게이트웨이를 VPC에 추가하여 VPN 연결을 설정합니다. 자세한 내용은 [VPC에 하드웨어 가상 프라이빗 게이트웨이 추가](#)를 참조하십시오.
2. ElastiCache 클러스터가 배포되는 서브넷의 VPC 라우팅 테이블을 업데이트하여 온-프레미스 애플리케이션 서버에서 들어오는 트래픽을 허용합니다. BGP를 사용하는 동적 라우팅의 경우 라우팅이 자동으로 전파될 수 있습니다.
3. ElastiCache 클러스터의 보안 그룹을 수정하여 온-프레미스 애플리케이션 서버에서 들어오는 인바운드 연결을 허용합니다.

VPN 연결을 통해 클러스터에 액세스하면 네트워크 지연 시간이 생기고 데이터 전송 비용이 추가로 발생합니다.

[Direct Connect를 사용하여 고객의 데이터 센터에서 실행되는 애플리케이션에서 ElastiCache 클러스터 액세스](#)

다음 다이어그램에서는 Direct Connect를 사용하여 회사 네트워크에서 실행되는 애플리케이션에서 ElastiCache 클러스터에 액세스하는 과정을 보여줍니다.



Direct Connect를 통해 데이터 센터에서 ElastiCache에 연결

Direct Connect를 사용하여 네트워크에서 실행되는 애플리케이션에서 ElastiCache 클러스터에 액세스하려면

1. Direct Connect 연결을 설정합니다. 자세한 내용은 [AWS Direct Connect 시작하기](#)를 참조하십시오.
2. ElastiCache 클러스터의 보안 그룹을 수정하여 온-프레미스 애플리케이션 서버에서 들어오는 인바운드 연결을 허용합니다.

DX 연결을 통해 클러스터에 액세스하면 네트워크 지연 시간이 생기고 데이터 전송 요금이 추가로 발생할 수 있습니다.

## Virtual Private Cloud(VPC) 생성

이 예제에서는 각 가용 영역에 대해 프라이빗 서브넷이 있는 Amazon VPC를 생성합니다.

### Amazon VPC 만들기 (콘솔)

Amazon Virtual Private Cloud내에서 ElastiCache 클러스터를 생성하려면

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/vpc/>에서 Amazon VPC 콘솔을 엽니다.
2. Amazon Virtual Private Cloud 마법사를 사용하여 새 Amazon VPC를 생성합니다.
  - a. 탐색 목록에서 [VPC Dashboard]를 선택합니다.
  - b. [Start VPC Wizard]를 선택합니다.

- c. Amazon VPC 마법사에서 퍼블릭 및 프라이빗 서브넷이 있는 VPC를 선택한 다음 다음을 선택합니다.
  - d. [VPC with Public and Private Subnets] 페이지에서 기본값을 유지하고 [Create VPC]를 선택합니다.
  - e. 나타나는 확인 메시지에서 [Close]를 선택합니다.
3. Amazon VPC에 두 개의 서브넷(퍼블릭 서브넷, 프라이빗 서브넷)이 있는지 확인합니다. 이러한 서브넷은 자동으로 생성됩니다.
    - a. 탐색 목록에서 [Subnets]를 선택합니다.
    - b. 서브넷 목록에서 Amazon VPC에 두 개의 서브넷이 있는지 찾아봅니다.

Subnet ID	State	VPC ID	Class of IP	Available IPs
subnet-5bf5e639	available	vpc-d3a77cb6	10.0.1.0/24 <b>Private</b>	251
subnet-58f5e63a	available	vpc-d3a77cb6	10.0.0.0/24 <b>Public</b>	250

마법사는 프라이빗 서브넷에서 인터넷으로의 아웃바운드 통신을 위해 Amazon EC2 NAT 인스턴스 및 탄력적 IP 주소(Amazon EC2 요금 적용)를 생성하기 때문에 퍼블릭 서브넷은 사용 가능한 IP 주소 수가 하나 더 적습니다.

**Tip**

서브넷 식별자 2개(퍼블릭 서브넷, 프라이빗 서브넷)를 기록해 둡니다. 추후 캐시 클러스터를 시작하고 Amazon VPC에 Amazon EC2 인스턴스를 추가할 때 이 정보가 필요합니다.

4. Amazon VPC 보안 그룹을 생성합니다. 캐시 클러스터 및 Amazon EC2 인스턴스에 이 그룹을 사용합니다.
  - a. Amazon VPC 관리 콘솔의 탐색 창에서 보안 그룹을 선택합니다.
  - b. [Create Security Group]을 선택합니다.
  - c. 보안 그룹의 이름과 설명을 해당 상자에 입력합니다. VPC 상자에서 Amazon VPC의 식별자를 선택합니다.

### Create Security Group Cancel

**Name:**

**Description:**

**VPC:**

---

- d. 원하는 대로 설정되었으면 [Yes, Create]를 선택합니다.

5. 보안 그룹의 네트워크 수신 규칙을 정의합니다. 이 규칙을 사용하면 SSH(Secure Shell)를 사용하여 Amazon EC2 인스턴스에 연결할 수 있습니다.
  - a. 탐색 목록에서 [Security Groups]를 선택합니다.
  - b. 목록에서 보안 그룹을 찾아 선택합니다.
  - c. [Security Group] 아래에서 [Inbound] 탭을 선택합니다. [Create a new rule] 상자에서 [SSH]를 선택한 다음 [Add Rule]을 선택합니다.
  - d. [Apply Rule Changes]를 선택합니다.

이제 캐시 서브넷 그룹을 생성하고 Amazon VPC에서 캐시 클러스터를 시작할 수 있습니다.

## 캐시 서브넷 그룹 생성

캐시 서브넷 그룹은 Amazon VPC에서 캐시 클러스터에 대해 지정할 수 있는 서브넷 모음입니다. Amazon VPC에서 캐시 클러스터를 시작할 때 캐시 서브넷 그룹을 선택해야 합니다. 그러면 ElastiCache가 해당 캐시 서브넷 그룹을 사용하여 해당 서브넷의 IP 주소를 클러스터의 각 캐시 노드에 할당합니다.

ElastiCache 관리 콘솔, AWS CLI 또는 ElastiCache API를 사용하여 서브넷 그룹을 생성하는 방법에 관한 지침은 [서브넷 그룹 생성 \(p. 183\)](#) 단원을 참조하십시오.

캐시 서브넷 그룹을 생성하면 Amazon VPC에서 실행할 캐시 클러스터를 시작할 수 있습니다. 다음 항목인 [Amazon VPC에서 캐시 클러스터 생성 \(p. 182\)](#) 단원으로 이동합니다.

## Amazon VPC에서 캐시 클러스터 생성

이 예제에서는 Amazon VPC에서 캐시 클러스터를 생성합니다.

### Amazon VPC (콘솔)에서 캐시 클러스터 생성

Memcached 캐시 클러스터를 시작하려면 [Memcached 클러스터 생성 \(콘솔\) \(p. 69\)](#)를 참조하십시오. 6.c단계에서 VPC 서브넷 그룹을 선택합니다.

Amazon VPC 내에서 캐시 클러스터를 시작했습니다. Amazon VPC에서 실행 중인 새 캐시 클러스터에 연결하는 방법의 예는 [Amazon VPC에서 실행되는 클러스터에 연결 \(p. 182\)](#) 단원을 참조하십시오.

## Amazon VPC에서 실행되는 클러스터에 연결

이 예제에서는 Amazon VPC에서 Amazon EC2 인스턴스를 시작하는 방법을 소개합니다. 이 인스턴스에 로그인하고 Amazon VPC에서 실행 중인 ElastiCache 클러스터를 액세스할 수 있습니다.

### Note

Amazon EC2 사용에 대한 자세한 내용은 [Amazon EC2 설명서의 Amazon EC2 Getting Started Guide](#)를 참조하십시오.

### Important

AWS 계정에 추가 요금이 발생하는 것을 방지하기 위해 이 예제를 실행한 후에 더 이상 필요하지 않은 AWS 리소스를 삭제하십시오.

클러스터 연결에 대한 자세한 내용은 ElastiCache 사용 설명서의 [3단계: 클러스터 노드에 연결 \(p. 16\)](#) 단원을 참조하십시오.

## 서브넷 및 서브넷 그룹

서브넷 그룹은 Amazon Virtual Private Cloud (VPC) 환경에서 실행 중인 클러스터에 대해 지정할 수 있는 서브넷(일반적으로 프라이빗 서브넷) 모음입니다.

Amazon VPC에서 클러스터를 생성할 경우 캐시 서브넷 그룹을 지정해야 합니다. ElastiCache는 해당 캐시 서브넷 그룹을 사용하여 캐시 노드에 연결된 서브넷 내의 서브넷 및 IP 주소를 선택합니다.

이 섹션에서는 서브넷과 서브넷 그룹을 생성 및 활용하여 ElastiCache 리소스에 대한 액세스를 관리하는 방법을 알아봅니다.

Amazon VPC 환경에서 서브넷 그룹 사용에 대한 자세한 내용은 [2단계: 액세스 자격 부여 \(p. 16\)](#) 단원을 참조하십시오.

주제

- [서브넷 그룹 생성 \(p. 183\)](#)
- [클러스터에 서브넷 그룹 지정 \(p. 185\)](#)
- [서브넷 그룹 수정 \(p. 185\)](#)
- [서브넷 그룹 삭제 \(p. 187\)](#)

## 서브넷 그룹 생성

새 서브넷 그룹을 생성할 때 사용 가능한 IP 주소의 수를 기록하십시오. 서브넷에 무료 IP 주소가 거의 없는 경우 클러스터에 추가할 수 있는 노드의 수가 제약될 수 있습니다. 이 문제를 해결하기 위해 클러스터의 가용 영역에 충분한 수의 IP 주소가 있도록 서브넷 그룹에 하나 이상의 서브넷을 지정할 수 있습니다. 그 이후 클러스터에 더 많은 노드를 추가할 수 있습니다.

다음 절차는 `mysubnetgroup` (console), AWS CLI 및 ElastiCache API라는 서브넷 그룹을 생성하는 방법을 보여줍니다.

### 서브넷 그룹 생성 (콘솔)

다음 절차는 서브넷 그룹 (console)을 생성하는 방법을 보여줍니다.

서브넷 그룹을 생성하려면 (콘솔)

1. AWS Management Console에 로그인한 후 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 목록에서 [Subnet Groups]를 선택합니다.
3. [Create Subnet Group]을 선택합니다.
4. [Create Subnet Group] 마법사에서 다음을 수행합니다. 모든 설정이 원하는 대로 설정되었으면 [Yes, Create]를 선택합니다.
  - a. [Name] 상자에 서브넷 그룹의 이름을 입력합니다.
  - b. [Description] 상자에 서브넷 그룹에 대한 설명을 입력합니다.
  - c. VPC ID 상자에서 생성한 Amazon VPC를 선택합니다.
  - d. [Availability Zone] 및 [Subnet ID] 목록에서 프라이빗 서브넷의 ID와 가용 영역을 선택한 다음 [Add]를 선택합니다.

5. 나타나는 확인 메시지에서 [Close]를 선택합니다.

새 서브넷 그룹이 ElastiCache 콘솔의 서브넷 그룹 목록에 나타납니다. 창 하단에서 서브넷 그룹을 선택하여 이 그룹과 연결된 모든 서브넷 등의 상세 내용을 확인할 수 있습니다.

## 서브넷 그룹 생성 (AWS CLI)

명령 프롬프트에서 `create-cache-subnet-group` 명령을 사용하여 서브넷 그룹을 생성합니다.

Linux, macOS, or Unix의 경우:

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

Windows의 경우:

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

이 명령은 다음과 유사한 출력을 생성합니다.

```
{
 "CacheSubnetGroup": {
 "VpcId": "vpc-37c3cd17",
 "CacheSubnetGroupDescription": "Testing",
 "Subnets": [
 {
 "SubnetIdentifier": "subnet-53df9c3a",
```

```
 "SubnetAvailabilityZone": {
 "Name": "us-west-2a"
 }
 },
],
 "CacheSubnetGroupName": "mysubnetgroup"
}
}
```

자세한 내용은 AWS CLI 항목 [create-cache-subnet-group](#)를 참조하십시오.

## 서브넷 그룹 생성 (ElastiCache API)

ElastiCache API를 사용하여 다음 파라미터와 함께 `CreateCacheSubnetGroup`을 호출합니다.

- `CacheSubnetGroupName=mysubnetgroup`
- `CacheSubnetGroupDescription=Testing`
- `SubnetIds.member.1=subnet-53df9c3a`

### Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=CreateCacheSubnetGroup
&CacheSubnetGroupDescription=Testing
&CacheSubnetGroupName=mysubnetgroup
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&SubnetIds.member.1=subnet-53df9c3a
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20141201T220302Z
&X-Amz-Expires=20141201T220302Z
&X-Amz-Signature=<signature>
&X-Amz-SignedHeaders=Host
```

## 클러스터에 서브넷 그룹 지정

서브넷 그룹을 생성한 후 Amazon VPC에서 클러스터를 시작할 수 있습니다. 자세한 내용은 다음을 참조하십시오.

- Memcached 클러스터 – 클러스터를 시작하려면 [Memcached 클러스터 생성 \(콘솔\)](#) (p. 69) 단원을 참조하십시오. 5.a 단계(고급 Memcached 설정)에서 VPC 서브넷 그룹을 선택합니다.

## 서브넷 그룹 수정

서브넷 그룹의 설명을 수정하거나 서브넷 그룹과 연관된 서브넷 ID의 목록을 수정할 수 있습니다. 클러스터가 현재 해당 서브넷을 사용하고 있는 경우 서브넷 그룹에서 서브넷 ID를 삭제할 수 없습니다.

다음 절차는 서브넷 그룹을 수정하는 방법을 보여줍니다.

### 서브넷 그룹 수정 (콘솔)

서브넷 그룹을 수정하려면

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.

2. 탐색 창에서 [Subnet Groups]를 선택합니다.
3. 서브넷 그룹 목록에서 수정하려는 서브넷 그룹을 선택합니다.
4. ElastiCache 콘솔의 하단부에서 설명 또는 서브넷 그룹에 대한 서브넷 ID의 목록을 변경합니다. 변경 사항을 저장하려면 [Save]를 선택합니다.

## 서브넷 그룹 수정 (AWS CLI)

명령 프롬프트에서 `modify-cache-subnet-group` 명령을 사용하여 서브넷 그룹을 수정합니다.

Linux, macOS, or Unix의 경우:

```
aws elasticache modify-cache-subnet-group \
 --cache-subnet-group-name mysubnetgroup \
 --cache-subnet-group-description "New description" \
 --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

Windows의 경우:

```
aws elasticache modify-cache-subnet-group ^
 --cache-subnet-group-name mysubnetgroup ^
 --cache-subnet-group-description "New description" ^
 --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

이 명령은 다음과 유사한 출력을 생성합니다.

```
{
 "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"
 }
}
```

자세한 내용은 AWS CLI 항목 [modify-cache-subnet-group](#)를 참조하십시오.

## 서브넷 그룹 수정 (ElastiCache API)

ElastiCache API를 사용하여 다음 파라미터와 함께 `ModifyCacheSubnetGroup`을 호출합니다.

- `CacheSubnetGroupName=mysubnetgroup`
- 변경할 값이 있는 다른 파라미터. 이 예제는 `CacheSubnetGroupDescription=New%20description`을 사용하여 서브넷 그룹의 설명을 변경합니다.

## Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=ModifyCacheSubnetGroup
&CacheSubnetGroupDescription=New%20description
&CacheSubnetGroupName=mysubnetgroup
&SubnetIds.member.1=subnet-42df9c3a
&SubnetIds.member.2=subnet-48fc21a9
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20141201T220302Z
&X-Amz-Expires=20141201T220302Z
&X-Amz-Signature=<signature>
&X-Amz-SignedHeaders=Host
```

## Note

새 서브넷 그룹을 생성할 때 사용 가능한 IP 주소의 수를 기록하십시오. 서브넷에 무료 IP 주소가 거의 없는 경우 클러스터에 추가할 수 있는 노드의 수가 제약될 수 있습니다. 이 문제를 해결하기 위해 클러스터의 가용 영역에 충분한 수의 IP 주소가 있도록 서브넷 그룹에 하나 이상의 서브넷을 지정할 수 있습니다. 그 이후 클러스터에 더 많은 노드를 추가할 수 있습니다.

## 서브넷 그룹 삭제

서브넷 그룹이 더 이상 필요하지 않다고 판단되면 삭제할 수 있습니다. 현재 클러스터에서 사용 중인 경우 서브넷 그룹을 삭제할 수 없습니다.

다음 절차는 서브넷 그룹을 삭제하는 방법을 보여줍니다.

### 서브넷 그룹 삭제 (콘솔)

서브넷 그룹을 삭제하는 방법

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 [Subnet Groups]를 선택합니다.
3. 서브넷 그룹 목록에서 삭제할 서브넷 그룹을 선택한 다음 [Delete]를 선택합니다.
4. 이 작업을 확인하라는 메시지가 표시되면 [Yes, Delete]를 선택합니다.

### 서브넷 그룹 삭제 (AWS CLI)

AWS CLI를 사용하여 다음 파라미터로 delete-cache-subnet-group 명령을 호출할 수 있습니다.

- --cache-subnet-group-name *mysubnetgroup*

Linux, macOS, or Unix의 경우:

```
aws elasticache delete-cache-subnet-group \
--cache-subnet-group-name mysubnetgroup
```

Windows의 경우:

```
aws elasticache delete-cache-subnet-group ^
--cache-subnet-group-name mysubnetgroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CLI 항목 [delete-cache-subnet-group](#)를 참조하십시오.

## 서브넷 그룹 삭제 (ElastiCache API)

ElastiCache API를 사용하여 다음 파라미터와 함께 DeleteCacheSubnetGroup을 호출합니다.

- CacheSubnetGroupName=*mysubnetgroup*

### Example

읽기 쉽도록 줄바꿈이 추가됩니다.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DeleteCacheSubnetGroup
&CacheSubnetGroupName=mysubnetgroup
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Timestamp=20141201T220302Z
&Version=2014-12-01
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20141201T220302Z
&X-Amz-Expires=20141201T220302Z
&X-Amz-Signature=<signature>
&X-Amz-SignedHeaders=Host
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 ElastiCache API 항목 [DeleteCacheSubnetGroup](#)을 참조하십시오.

## 보안 그룹: EC2-Classic

### Important

Amazon ElastiCache 보안 그룹은 Amazon Virtual Private Cloud 환경(VPC)에서 실행되지 않는 클러스터에만 적용됩니다. Amazon Virtual Private Cloud를 실행하는 경우 콘솔 탐색 창에서 보안 그룹을 사용할 수 없습니다.

Amazon VPC에서 ElastiCache 노드를 실행하는 경우, ElastiCache 보안 그룹과 다른 Amazon VPC 보안 그룹으로 클러스터에 대한 액세스를 제어합니다. Amazon VPC에서 ElastiCache를 사용하는 방법에 대한 자세한 내용은 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

Amazon ElastiCache에서는 ElastiCache 보안 그룹을 통해 클러스터에 대한 액세스를 제어할 수 있습니다. ElastiCache 보안 그룹은 클러스터에 대한 네트워크 액세스를 제어하는 방화벽 역할을 합니다. 기본적으로 네트워크 액세스는 클러스터에서 해제됩니다. 애플리케이션이 클러스터에 액세스하도록 하려면 특정 Amazon EC2 보안 그룹의 호스트에서 명시적으로 액세스를 활성화해야 합니다. 수신 규칙이 구성되면 동일한 규칙이 해당 보안 그룹과 관련된 모든 클러스터에 적용됩니다.

클러스터에 대한 네트워크 액세스를 허용하려면, 보안 그룹을 생성하고 AuthorizeCacheSecurityGroupIngress API 작업(CLI: `authorize-cache-security-group-ingress`)을 사용하여 원하는 Amazon EC2 보안 그룹에 권한을 부여합니다(그런 다음 허용된 Amazon EC2 인스턴스를 지정). 보안 그룹은 생성 시 또는 ModifyCacheCluster API 작업(CLI: `modify-cache-cluster`)을 사용하여 클러스터에 연결할 수 있습니다.

## Important

IP 범위 기반 액세스 제어는 현재 개별 클러스터 수준에서 비활성화되어 있습니다. 클러스터에 액세스하는 모든 클라이언트는 EC2 네트워크 내에 있어야 하며, 이전에 설명한 대로 보안 그룹을 통해 권한을 부여받아야 합니다.

Amazon VPC에서 ElastiCache를 사용하는 방법에 대한 자세한 내용은 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

Amazon VPC에서 실행되는 Amazon EC2 인스턴스는 EC2-Classical의 ElastiCache 클러스터에 연결할 수 없습니다.

### 주제

- [보안 그룹 생성 \(p. 189\)](#)
- [사용 가능한 보안 그룹 나열 \(p. 190\)](#)
- [보안 그룹 보기 \(p. 191\)](#)
- [Amazon EC2 보안 그룹에 대한 네트워크 액세스 권한 부여 \(p. 193\)](#)

## 보안 그룹 생성

이 주제는 Amazon VPC에서 실행하지 않는 경우에만 관련이 있습니다. Amazon VPC에서 실행하는 경우 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

보안 그룹을 생성하려면 이름과 설명을 제공해야 합니다.

다음 절차는 새 보안 그룹을 생성하는 방법을 보여줍니다.

### 보안 그룹 생성 (콘솔)

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. [Create Security Group]을 선택합니다.
4. [Create Security Group]의 [Security Group]에 새 보안 그룹의 이름을 입력합니다.
5. [Description]에 새 보안 그룹에 대한 설명을 입력합니다.
6. Create를 선택합니다.

### 보안 그룹 생성 (AWS CLI)

명령 프롬프트에서 다음 파라미터와 함께 `create-cache-security-group` 명령을 사용합니다.

- `--cache-security-group-name` – 생성 중인 보안 그룹의 이름입니다.

예: `mysecuritygroup`

- `--description` – 이 보안 그룹에 대한 설명입니다.

예: "나의 새 보안 그룹"

Linux, macOS, or Unix의 경우:

```
aws elasticache create-cache-security-group \
 --cache-security-group-name mysecuritygroup \
 --description "나의 새 보안 그룹"
```

```
--description "My new security group"
```

Windows의 경우:

```
aws elasticache create-cache-security-group ^
 --cache-security-group-name mysecuritygroup ^
 --description "My new security group"
```

자세한 내용은 [create-cache-security-group](#) 단원을 참조하십시오.

## 보안 그룹 생성 (ElastiCache API)

ElastiCache API 작업 `CreateCacheSecurityGroup`을 다음 파라미터와 함께 사용:

- `CacheSecurityGroupName` – 생성 중인 보안 그룹의 이름입니다.

예: mysecuritygroup

- `Description` – 이 보안 그룹에 대한 URL로 인코딩된 설명입니다.

예: My%20security%20group

### Example

읽기 쉽도록 줄바꿈이 추가됩니다.

```
https://elasticache.us-west-2.amazonaws.com /
?Action=CreateCacheSecurityGroup
&CacheSecurityGroupName=mysecuritygroup
&Description=My%20security%20group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T220302Z
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20150202T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20150202T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

## 사용 가능한 보안 그룹 나열

이 주제는 Amazon VPC에서 실행하지 않는 경우에만 관련이 있습니다. Amazon VPC에서 실행하는 경우 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

AWS 계정에 대해 생성된 보안 그룹을 나열할 수 있습니다.

다음 절차는 AWS 계정에 대해 사용할 수 있는 보안 그룹을 나열하는 방법을 보여줍니다.

### 사용 가능한 보안 그룹 나열 (콘솔)

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.

사용 가능한 보안 그룹이 [ Security Groups] 목록에 나타납니다.

## 사용 가능한 보안 그룹 나열 (AWS CLI)

명령 프롬프트에서 `describe-cache-security-groups` 명령을 사용하여 AWS 계정에 사용할 수 있는 모든 보안 그룹을 나열합니다.

```
aws elasticache describe-cache-security-groups
```

이 명령의 JSON 출력은 다음과 같습니다.

```
{
 "Marker": "Marker",
 "CacheSecurityGroups": [
 {
 "OwnerId": "OwnerId",
 "CacheSecurityGroupName": "CacheSecurityGroupName",
 "Description": "Description",
 "EC2SecurityGroups": [
 {
 "Status": "Status",
 "EC2SecurityGroupName": "EC2SecurityGroupName",
 "EC2SecurityGroupOwnerId": "EC2SecurityGroupOwnerId"
 }
]
 }
]
}
```

자세한 내용은 [describe-cache-security-groups](#)을 참조하십시오.

## 사용 가능한 보안 그룹 나열 (ElastiCache API)

ElastiCache API를 사용하여 `DescribeCacheSecurityGroups`를 호출합니다.

### Example

읽기 쉽도록 줄바꿈이 추가됩니다.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheSecurityGroups
&MaxRecords=100
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T220302Z
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20150202T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20150202T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

## 보안 그룹 보기

이 주제는 Amazon VPC에서 실행하지 않는 경우에만 관련이 있습니다. Amazon VPC에서 실행하는 경우 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

보안 그룹에 대한 세부 정보를 볼 수 있습니다.

다음 절차는 ElastiCache 콘솔, AWS CLI 및 ElastiCache API를 사용하여 보안 그룹의 속성을 보는 방법을 보여줍니다.

## 보안 그룹 보기 (콘솔)

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.

사용 가능한 캐시 보안 그룹이 [Security Groups] 목록에 나타납니다.

3. [Security Groups] 목록에서 캐시 보안 그룹을 선택합니다.

보안 그룹에 대해 정의된 권한 부여의 목록이 창 하단의 세부 정보 단원에 표시됩니다.

## 보안 그룹 보기 (AWS CLI)

명령 프롬프트에서 보려는 보안 그룹의 이름과 함께 AWS CLI `describe-cache-security-groups` 명령을 사용합니다.

- `--cache-security-group-name` - 세부 정보가 반환되는 보안 그룹의 이름입니다.

```
aws elasticache describe-cache-security-groups --cache-security-group-name mysecuritygroup
```

이 명령의 JSON 출력은 다음과 같습니다.

```
{
 "CacheSecurityGroup": {
 "OwnerId": "OwnerId",
 "CacheSecurityGroupName": "CacheSecurityGroupName",
 "Description": "Description",
 "EC2SecurityGroups": [
 {
 "Status": "Status",
 "EC2SecurityGroupName": "EC2SecurityGroupName",
 "EC2SecurityGroupOwnerId": "EC2SecurityGroupOwnerId"
 }
]
 }
}
```

자세한 내용은 `describe-cache-security-groups` 단원을 참조하십시오.

## 보안 그룹 보기 (ElastiCache API)

ElastiCache API를 사용하여 보려는 보안 그룹의 이름으로 `DescribeCacheSecurityGroups`를 호출합니다.

- `CacheSecurityGroupName` - 세부 정보가 반환되는 캐시 보안 그룹의 이름입니다.

### Example

읽기 쉽도록 줄바꿈이 추가됩니다.

```
https://elasticache.amazonaws.com/
?Action=DescribeCacheSecurityGroups
&CacheSecurityGroupName=mysecuritygroup
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
```

```
&Timestamp=20150202T220302Z
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20150202T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20150202T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

## Amazon EC2 보안 그룹에 대한 네트워크 액세스 권한 부여

이 주제는 Amazon VPC에서 실행하지 않는 경우에만 관련이 있습니다. Amazon VPC에서 실행하는 경우 [Amazon VPC 및 ElastiCache 보안 \(p. 170\)](#) 단원을 참조하십시오.

Amazon EC2 인스턴스에서 클러스터에 액세스하려면 EC2 인스턴스가 속한 Amazon EC2 보안 그룹에 대한 액세스 권한을 부여해야 합니다. 다음 절차는 Amazon EC2 보안 그룹에 대한 액세스 권한을 부여하는 방법을 보여줍니다.

### Important

- Amazon EC2 보안 그룹에 대한 권한 부여는 Amazon EC2 보안 그룹에 속한 모든 EC2 인스턴스의 클러스터에 대한 액세스 권한만 부여합니다.
- 액세스 권한 변경이 적용되는 데는 약 1분이 걸립니다.

## Amazon EC2 보안 그룹에 대한 네트워크 액세스 권한 부여(콘솔)

1. AWS Management 콘솔에 로그인하고 <https://console.aws.amazon.com/elasticache/>에서 ElastiCache 콘솔을 엽니다.
2. 탐색 창에서 [Security Groups]를 선택합니다.
3. [Security Groups] 목록에서 액세스 권한을 부여할 보안 그룹의 왼쪽에 있는 상자를 선택합니다.
4. 창 하단에 있는 [EC2 Security Group Name] 목록에서 Amazon EC2 보안 그룹을 선택합니다.
5. [추가]를 선택합니다.

## Amazon EC2 보안 그룹에 대한 네트워크 액세스 권한 부여(AWS CLI)

명령 프롬프트에서 다음 파라미터와 함께 `authorize-cache-security-group-ingress` 명령을 사용하여 Amazon EC2 보안 그룹에 대한 액세스 권한을 부여합니다.

- `--cache-security-group-name` – Amazon EC2 액세스 권한을 부여할 보안 그룹의 이름입니다.
- `--ec2-security-group-name` – Amazon EC2 인스턴스가 속한 Amazon EC2 보안 그룹의 이름입니다.
- `--ec2-security-group-owner-id` – Amazon EC2 보안 그룹의 소유자 ID입니다.

### Example

Linux, macOS, or Unix의 경우:

```
aws elasticache authorize-cache-security-group-ingress \
 --cache-security-group-name default \
 --ec2-security-group-name myec2group \
 --ec2-security-group-owner-id 987654321021
```

Windows의 경우:

```
aws elasticache authorize-cache-security-group-ingress ^
```

```
--cache-security-group-name default ^
--ec2-security-group-name myec2group ^
--ec2-security-group-owner-id 987654321021
```

다음과 비슷한 출력이 생성되어야 합니다.

```
{
 "CacheSecurityGroup": {
 "OwnerId": "OwnerId",
 "CacheSecurityGroupName": "CacheSecurityGroupName",
 "Description": "Description",
 "EC2SecurityGroups": [
 {
 "Status": "available",
 "EC2SecurityGroupName": "EC2SecurityGroupName",
 "EC2SecurityGroupOwnerId": "EC2SecurityGroupOwnerId"
 }
]
 }
}
```

자세한 내용은 [authorize-cache-security-group-ingress](#) 단원을 참조하십시오.

## Amazon EC2 보안 그룹에 대한 네트워크 액세스 권한 부여(ElastiCache API)

ElastiCache API를 사용하여 다음 파라미터와 함께 `AuthorizeCacheSecurityGroupIngress`을 호출합니다.

- `CacheSecurityGroupName` – Amazon EC2 액세스 권한을 부여할 보안 그룹의 이름입니다.
- `EC2SecurityGroupName` – Amazon EC2 인스턴스가 속한 Amazon EC2 보안 그룹의 이름입니다.
- `EC2SecurityGroupOwnerId` – Amazon EC2 보안 그룹의 소유자 ID입니다.

### Example

```
https://elasticache.us-west-2.amazonaws.com/
?Action=AuthorizeCacheSecurityGroupIngress
&EC2SecurityGroupOwnerId=987654321021
&EC2SecurityGroupName=myec2group
&Version=2015-02-02
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20150202T220302Z
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Date=20150202T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20150202T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

자세한 내용은 [AuthorizeCacheSecurityGroupIngress](#) 단원을 참조하십시오.

# Amazon ElastiCache의 Identity and Access Management(IAM)

Amazon ElastiCache에 액세스하려면 AWS가 요청을 인증하는 데 사용할 수 있는 자격 증명이 필요합니다. 이러한 자격 증명에는 ElastiCache 캐시 클러스터 또는 Amazon Elastic Compute Cloud (Amazon EC2)

인스턴스 같은 AWS 리소스에 액세스할 수 있는 권한이 있어야 합니다. 다음 단원에서는 리소스에 액세스할 수 있는지 대상을 제어하여 리소스를 보호할 수 있도록 [AWS Identity and Access Management\(IAM\)](#) 및 ElastiCache를 사용하는 방법에 대한 세부 정보를 제공합니다.

- [인증 \(p. 195\)](#)
- [액세스 제어 \(p. 196\)](#)

## 인증

다음과 같은 ID 유형으로 AWS에 액세스할 수 있습니다.

- **AWS 계정 루트 사용자** – AWS 계정을 처음 생성할 때는 해당 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 SSO(Single Sign-In) ID로 시작합니다. 이 자격 증명은 AWS 계정 루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업은 물론 관리 작업에도 루트 사용자를 사용하지 않는 것이 좋습니다. 대신 [IAM 사용자를 처음 생성할 때만 루트 사용자를 사용하는 모범 사례](#)를 준수하십시오. 그런 다음 루트 사용자 자격 증명을 안전하게 보관해 두고 몇 가지 계정 및 서비스 관리 작업을 수행할 때만 해당 자격 증명을 사용합니다.
- **IAM 사용자** – [IAM 사용자](#)는 특정 사용자 지정 권한(예: ElastiCache에서 a cluster을 생성할 권한)이 있는 AWS 계정 내 자격 증명입니다. IAM 사용자 이름과 암호를 사용하여 [AWS Management 콘솔](#), [AWS 토론 포럼](#) 또는 [AWS Support Center](#) 같은 보안 AWS 웹 페이지에 로그인할 수 있습니다.

사용자 이름과 암호 외에도 각 사용자에 대해 [액세스 키](#)를 생성할 수 있습니다. [여러 SDK 중 하나](#)를 통해 또는 [AWS Command Line Interface\(CLI\)](#)를 사용하여 AWS 제품에 프로그래밍 방식으로 액세스할 때 이러한 키를 사용할 수 있습니다. SDK 및 CLI 도구는 액세스 키를 사용하여 암호화 방식으로 요청에 서명합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. ElastiCache supports는 인바운드 API 요청을 인증하기 위한 프로토콜인 서명 버전 4를 지원합니다. 요청 인증에 대한 자세한 내용은 AWS General Reference의 [서명 버전 4 서명 프로세스](#) 단원을 참조하십시오.

- **IAM 역할** – [IAM 역할](#)은 특정 권한을 가진 계정에서 생성할 수 있는 IAM 자격 증명입니다. IAM 역할은 AWS에서 자격 증명으로 할 수 있는 것과 할 수 없는 것을 결정하는 권한 정책을 포함하는 AWS 자격 증명이라는 점에서 IAM 사용자와 유사합니다. 그러나 역할은 한 사람과만 연관되지 않고 해당 역할이 필요한 사람이라면 누구든지 맡을 수 있어야 합니다. 또한 역할에는 그와 연관된 암호 또는 액세스 키와 같은 표준 장기 자격 증명도 없습니다. 대신에 역할을 수임한 사람에게는 해당 역할 세션을 위한 임시 보안 자격 증명도 제공됩니다. 임시 자격 증명에 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.
- **연합된 사용자 액세스** – IAM 사용자를 생성하는 대신 AWS Directory Service의 기존 ID, 엔터프라이즈 사용자 디렉터리 또는 웹 ID 공급자를 사용할 수 있습니다. 이 사용자를 연합된 사용자라고 합니다. AWS에서는 [ID 공급자](#)를 통해 액세스가 요청되면 연합된 사용자에게 역할을 할당합니다. 연합된 사용자에 대한 자세한 내용은 IAM 사용 설명서의 [연합된 사용자 및 역할](#)을 참조하십시오.
- **AWS 서비스 액세스** – 서비스 역할은 서비스가 사용자를 대신하여 사용자 계정에서 작업을 수행하기 위해 수임하는 IAM 역할입니다. 일부 AWS 서비스 환경을 설정할 때 서비스에서 맡을 역할을 정의해야 합니다. 이 서비스 역할에는 서비스가 AWS 리소스에 액세스하는 데 필요한 모든 권한이 포함되어야 합니다. 서비스 역할은 서비스마다 다르지만 해당 서비스에 대한 문서화된 요구 사항을 충족하는 한 대부분의 경우 권한을 선택할 수 있습니다. 서비스 역할은 해당 계정 내 액세스 권한만 제공하며 다른 계정의 서비스에 대한 액세스 권한을 부여하는 데 사용될 수 없습니다. IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 예를 들어 Amazon Redshift에서 사용자 대신 Amazon S3 버킷에 액세스하도록 허용하는 역할을 생성한 후 해당 버킷에 있는 데이터를 Amazon Redshift 클러스터로 로드할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [AWS 서비스에 대한 권한을 위임할 역할 생성](#)을 참조하십시오.

- Amazon EC2에서 실행 중인 애플리케이션 – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 자격 증명을 얻을 수 있습니다. 자세한 내용은 IAM 사용 설명서의 [IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여](#)를 참조하십시오.

## 액세스 제어

요청을 인증할 수 있는 유효한 자격 증명이라도 권한이 없다면 Amazon ElastiCache 리소스를 생성하거나 액세스할 수 없습니다. 예를 들어, ElastiCache 캐시 클러스터를 생성할 권한이 있어야 합니다.

다음 단원에서는 Amazon ElastiCache에 대한 권한을 관리하는 방법을 설명합니다. 먼저 개요를 읽어 보면 도움이 됩니다.

- [ElastiCache 리소스에 대한 액세스 권한 관리 개요 \(p. 196\)](#)
- [Amazon ElastiCache에 대한 자격 증명 기반 정책\(IAM 정책\) 사용 \(p. 199\)](#)

## ElastiCache 리소스에 대한 액세스 권한 관리 개요

모든 AWS 리소스는 AWS 계정의 소유이고, 리소스 생성 또는 리소스 액세스 권한은 권한 정책에 따라 결정됩니다. 계정 관리자는 IAM 자격 증명(사용자, 그룹 및 역할)에 권한 정책을 연결할 수 있습니다. 또한, AWS Lambda 같은 일부 서비스에서도 권한 정책을 리소스에 연결할 수 있습니다.

### Note

계정 관리자 또는 관리자 사용자는 관리자 권한이 있는 사용자입니다. 자세한 정보는 IAM 사용 설명서에서 [IAM 모범 사례](#)를 참조하십시오.

사용 권한을 부여할 때 누구에게 권한을 부여할지 결정합니다. 또한 권한 대상이 되는 리소스와 해당 리소스에 허용되는 특정 작업을 결정합니다.

### 주제

- [Amazon ElastiCache 리소스 및 작업 \(p. 196\)](#)
- [리소스 소유권 이해 \(p. 197\)](#)
- [리소스 액세스 관리 \(p. 197\)](#)
- [정책 요소 지정: 작업, 효과, 리소스, 보안 주체 \(p. 198\)](#)
- [정책에서 조건 지정 \(p. 199\)](#)

## Amazon ElastiCache 리소스 및 작업

Amazon ElastiCache에서는 캐시 클러스터가 기본 리소스입니다.

다음 표에 나와 있는 것처럼 이러한 리소스에는 고유한 Amazon 리소스 이름(ARN)이 연결됩니다.

리소스 유형	ARN 형식
캐시 클러스터	<code>arn:aws:elasticache:<i>region</i>:<i>account-id</i>:cluster:<i>resource-name</i></code>

ElastiCache의 경우 ElastiCache 리소스를 처리하기 위한 작업을 제공합니다. 사용 가능한 작업 목록은 Amazon ElastiCache [작업](#)을 참조하십시오.

## 리소스 소유권 이해

리소스 소유자는 해당 리소스를 만든 AWS 계정입니다. 즉, 리소스 소유자는 리소스를 생성하는 요청을 인증하는 보안 주체 엔터티의 AWS 계정입니다. 보안주체 엔터티는 루트 계정, IAM 사용자 또는 IAM 역할일 수 있습니다. 다음 예에서는 이 계정의 작동 방식을 설명합니다.

- AWS 계정의 루트 계정 자격 증명을 사용하여 캐시 클러스터를 생성한다고 가정합니다. 이 경우, AWS 계정이 리소스의 소유자입니다. ElastiCache에서 리소스는 캐시 클러스터입니다.
- AWS 계정에서 IAM 사용자를 생성하고 해당 사용자에게 캐시 클러스터 생성 권한을 부여한다고 가정합니다. 이러한 경우, 이 사용자가 캐시 클러스터를 생성할 수 있습니다. 하지만 캐시 클러스터 리소스는 사용자가 속한 AWS 계정의 소유입니다.
- AWS 계정에서 IAM 역할을 생성하고 해당 역할에게 캐시 클러스터 생성 권한을 부여한다고 가정합니다. 이러한 경우, 이 역할을 수임할 사람은 캐시 클러스터를 생성할 수 있습니다. 캐시 클러스터 리소스는 역할이 속한 AWS 계정의 소유입니다.

## 리소스 액세스 관리

권한 정책은 누가 무엇에 액세스 할 수 있는지를 나타냅니다. 다음 단원에서는 권한 정책을 만드는 데 사용할 수 있는 옵션에 대해 설명합니다.

### Note

이 단원에서는 Amazon ElastiCache의 맥락에서 IAM을 사용하는 방법에 대해 설명하며, IAM 서비스에 대한 자세한 내용은 다루지 않습니다. 전체 IAM 설명서는 IAM 사용 설명서의 [IAM이란 무엇인가?](#)를 참조하십시오. IAM 정책 구분과 설명에 대한 자세한 정보는 IAM 사용 설명서의 [AWS IAM 정책 참조](#)를 참조하십시오.

IAM 자격 증명에 연결된 정책은 자격 증명 기반 정책(IAM 정책)이라고 합니다. 리소스에 연결된 정책은 리소스 기반 정책이라고 합니다. Amazon ElastiCache는 자격 증명 기반 정책(IAM 정책)만 지원합니다.

### 주제

- [자격 증명 기반 정책\(IAM 정책\) \(p. 197\)](#)
- [리소스 기반 정책 \(p. 198\)](#)

## 자격 증명 기반 정책(IAM 정책)

정책을 IAM 자격 증명에 연결할 수 있습니다. 예를 들면,

- 계정 내 사용자 또는 그룹에 권한 정책 연결 – 계정 관리자는 권한을 부여하기 위해 특정 사용자에게 연결된 권한 정책을 사용할 수 있습니다. 이 경우 해당 사용자는 캐시 클러스터, 파라미터 그룹 또는 보안 그룹과 같은 ElastiCache 리소스를 생성할 수 있습니다.
- 역할에 권한 정책 연결(교차 계정 권한 부여) – 자격 증명 기반 권한 정책을 IAM 역할에 연결하여 교차 계정 권한을 부여할 수 있습니다. 예를 들어, 계정 A의 관리자는 다음과 같이 다른 AWS 계정(예: 계정 B) 또는 AWS 서비스에 교차 계정 권한을 부여할 역할을 생성할 수 있습니다.
  1. 계정 A 관리자는 IAM 역할을 생성하고 계정 A의 리소스에 대한 권한을 부여하는 역할에 권한 정책을 연결합니다.
  2. 계정 A 관리자는 계정 B를 역할 수임할 보안 주체로 식별하는 역할에 신뢰 정책을 연결합니다.
  3. 계정 B 관리자는 계정 B의 사용자에게 역할을 수임할 권한을 위임할 수 있습니다. 그러면 계정 B의 사용자가 계정 A에서 리소스를 생성하거나 액세스할 수 있습니다. 일부 경우에는, 역할에 할당할 AWS 서비스 권한을 부여하려고 할 수 있습니다. 이러한 접근 방식을 지원하려면, 신뢰 정책의 보안 주체는 AWS 서비스 보안 주체일 수도 있습니다.

IAM을 사용하여 권한을 위임하는 방법에 대한 자세한 정보는 IAM 사용 설명서의 [액세스 관리](#)를 참조하십시오.

다음은 사용자가 ASW 계정에 대해 DescribeCacheClusters 작업을 수행할 수 있도록 허용하는 정책에 예제입니다. 현재 구현에서 ElastiCache에서는 API 작업에 대한 리소스 ARN을 사용하여 특정 리소스를 식별할 수 있도록 지원하지 않습니다. (이러한 접근 방식을 리소스 수준 권한이라고도 합니다.) 따라서 와일드카드 문자(\*)를 지정해야 합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "DescribeCacheClusters",
 "Effect": "Allow",
 "Action": [
 "elasticache:DescribeCacheClusters"],
 "Resource": "*"
 }
]
```

ElastiCache에서 자격 증명 기반 정책을 사용하는 방법에 대한 자세한 내용은 [Amazon ElastiCache에 대한 자격 증명 기반 정책\(IAM 정책\) 사용 \(p. 199\)](#) 단원을 참조하십시오. 사용자, 그룹, 역할 및 권한에 대한 자세한 내용은 IAM 사용 설명서의 [자격 증명\(사용자, 그룹 및 역할\)](#)을 참조하십시오.

## 리소스 기반 정책

Amazon S3와 같은 다른 서비스도 리소스 기반 권한 정책을 지원합니다. 예를 들어, 정책을 S3 버킷에 연결하여 해당 버킷에 대한 액세스 권한을 관리할 수 있습니다. Amazon ElastiCache는 리소스 기반 정책을 지원하지 않습니다.

## 정책 요소 지정: 작업, 효과, 리소스, 보안 주체

각 Amazon ElastiCache 리소스([Amazon ElastiCache 리소스 및 작업 \(p. 196\)](#) 참조)에 대해 서비스는 API 작업을 정의합니다([작업](#) 참조). 이러한 API 작업에 대한 권한을 부여하기 위해 ElastiCache에서는 정책에서 지정할 수 있는 작업을 정의합니다. 예를 들어 ElastiCache 스냅샷 리소스에 대해 CreateCacheCluster, DeleteCacheCluster 및 DescribeCacheCluster 작업을 정의합니다. API 작업을 실시하려면 둘 이상의 작업에 대한 권한이 필요할 수 있습니다.

다음은 가장 기본적인 정책 요소입니다.

- 리소스 – 정책에서 Amazon 리소스 이름(ARN)을 사용하여 정책을 적용할 리소스를 식별합니다. ElastiCache 리소스의 경우에는 IAM 정책에 항상 와일드카드 문자(\*)를 사용합니다. 자세한 내용은 [Amazon ElastiCache 리소스 및 작업 \(p. 196\)](#) 단원을 참조하십시오.
- 작업 – 작업 키워드를 사용하여 허용 또는 거부할 리소스 작업을 식별합니다. 예를 들어 지정한 Effect에 따라 elasticache:CreateCacheCluster 권한은 Amazon ElastiCache CreateCacheCluster 작업을 수행할 수 있는 사용자 권한을 허용하거나 거부합니다.
- 결과 – 사용자가 특정 작업을 요청하는 경우의 결과를 지정합니다. 이는 허용 또는 거부 중에 하나가 될 수 있습니다. 명시적으로 리소스에 대한 액세스 권한을 부여(허용)하지 않는 경우, 액세스는 묵시적으로 거부됩니다. 리소스에 대한 액세스를 명시적으로 거부할 수도 있습니다. 예를 들어, 다른 정책에서 액세스 권한을 부여하더라도 사용자가 해당 리소스에 액세스할 수 없도록 하려고 할 때 이러한 작업을 수행할 수 있습니다.
- 보안 주체 – 자격 증명 기반 정책(IAM 정책)에서 정책이 연결되는 사용자는 암시적인 보안 주체입니다. 리소스 기반 정책의 경우 사용자, 계정, 서비스 또는 권한의 수신자인 기타 개체를 지정합니다(리소스 기반 정책에만 해당). ElastiCache의 경우 리소스 기반 정책을 지원하지 않습니다.

IAM 정책 구문과 설명에 대한 자세한 내용은 IAM 사용 설명서의 [AWS IAM 정책 참조](#)를 참조하십시오.

모든 Amazon ElastiCache API 작업을 보여 주는 표는 [ElastiCache API 권한: 작업, 리소스 및 조건 참조 \(p. 209\)](#) 단원을 참조하십시오.

## 정책에서 조건 지정

권한을 부여할 때 IAM 정책 언어를 사용하여 정책이 적용되는 조건을 지정할 수 있습니다. 예를 들어, 특정 날짜 이후에만 정책을 적용할 수 있습니다. 정책 언어에서의 조건 지정에 관한 자세한 내용은 IAM 사용 설명서의 [조건](#) 단원을 참조하십시오.

조건을 표시하려면 미리 정의된 조건 키를 사용합니다. Amazon ElastiCache에만 해당되는 특정한 조건 키는 없습니다. 하지만 적절하게 사용할 수 있는 AWS 차원의 조건 키가 있습니다. AWS 차원 키의 전체 목록은 IAM 사용 설명서의 [사용 가능한 조건 키](#) 단원을 참조하십시오.

## Amazon ElastiCache에 대한 자격 증명 기반 정책(IAM 정책) 사용

이 단원에서는 계정 관리자가 IAM 자격 증명(즉 사용자, 그룹, 역할)에 권한 정책을 연결할 수 있는 자격 증명 기반 정책의 예를 제시합니다.

### Important

Amazon ElastiCache 리소스 액세스를 관리하기 위한 기본 개념과 옵션을 설명하는 주제를 먼저 읽어 보는 것이 좋습니다. 자세한 내용은 [ElastiCache 리소스에 대한 액세스 권한 관리 개요 \(p. 196\)](#) 단원을 참조하십시오.

이 주제의 섹션에서는 다음 내용을 학습합니다.

- [Amazon ElastiCache 콘솔 사용에 필요한 권한 \(p. 200\)](#)
- [Amazon ElastiCache에 대한 AWS 관리형\(미리 정의된\) 정책 \(p. 201\)](#)
- [고객 관리형 정책 예 \(p. 201\)](#)

다음은 권한 정책의 예입니다.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "AllowClusterPermissions",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheCluster",
 "elasticache:CreateReplicationGroup",
 "elasticache:DescribeCacheClusters",
 "elasticache:ModifyCacheCluster",
 "elasticache:RebootCacheCluster"],
 "Resource": "*"
 },
 {
 "Sid": "AllowUserToPassRole",
 "Effect": "Allow",
 "Action": ["iam:PassRole"],
 "Resource": "*"
 }
]
}
```

이 정책에는 두 명령문이 있습니다:

- 첫 번째 명령문은 계정이 소유하는 임의의 캐시 클러스터에서 Amazon ElastiCache 작업 (elasticache:CreateCacheCluster, elasticache:DescribeCacheClusters, elasticache:ModifyCacheCluster 및 elasticache:RebootCacheCluster)에 대한 권한을 부여합니다. 현재 Amazon ElastiCache는 리소스 수준의 작업에 대한 권한을 지원하지 않습니다. 따라서 정책은 와일드카드 문자(\*)를 Resource 값으로 지정합니다.
- 두 번째 명령문은 IAM 역할에서 IAM 작업(iam:PassRole)에 대한 권한을 부여합니다. Resource 값의 끝에 있는 와일드카드 문자(\*)는 명령문이 모든 IAM 역할에서 iam:PassRole 작업에 대한 권한을 허용한다는 의미입니다. 이러한 권한을 특정 역할로 제한하려면 리소스 ARN에 있는 와일드카드 문자(\*)를 특정 역할 이름으로 대체합니다.

자격 증명 기반 정책에서는 권한을 가질 보안 주체를 지정하지 않으므로 이 정책은 Principal 요소를 지정하지 않습니다. 정책을 사용자에게 연결할 경우, 사용자는 묵시적인 보안 주체가 됩니다. IAM 역할에 권한 정책을 연결하면 역할의 신뢰 정책에서 식별된 보안 주체가 권한을 얻습니다.

모든 Amazon ElastiCache API 작업과 해당 작업이 적용되는 리소스를 보여주는 표는 [ElastiCache API 권한: 작업, 리소스 및 조건 참조 \(p. 209\)](#) 단원을 참조하십시오.

## Amazon ElastiCache 콘솔 사용에 필요한 권한

권한 참조 표에 Amazon ElastiCache API 작업과 각 작업에 필요한 권한이 나열되어 있습니다. ElastiCache API 작업에 대한 자세한 내용은 [ElastiCache API 권한: 작업, 리소스 및 조건 참조 \(p. 209\)](#) 단원을 참조하십시오.

Amazon ElastiCache 콘솔을 사용하려면 먼저 다음 권한 정책과 같이 추가 작업에 대한 권한을 부여해야 합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "MinPermsForECConsole",
 "Effect": "Allow",
 "Action": [
 "elasticache:Describe*",
 "elasticache:List*",
 "ec2:DescribeAvailabilityZones",
 "ec2:DescribeVpcs",
 "ec2:DescribeAccountAttributes",
 "ec2:DescribeSecurityGroups",
 "cloudwatch:GetMetricStatistics",
 "cloudwatch:DescribeAlarms",
 "s3:ListAllMyBuckets",
 "sns:ListTopics",
 "sns:ListSubscriptions"],
 "Resource": "*"
 }
]
```

ElastiCache 콘솔에 이러한 추가 권한이 필요한 이유는 다음과 같습니다.

- ElastiCache 작업 권한이 있으면 콘솔에서 계정의 ElastiCache 리소스를 표시할 수 있습니다.
- 콘솔에 Amazon EC2 쿼리를 위한 ec2 작업 권한이 있어야 합니다. 그래야 가용 영역, VPC, 보안 그룹 및 계정 속성을 표시할 수 있습니다.
- cloudwatch 작업에 대한 권한이 있으면 콘솔에서 Amazon CloudWatch 지표와 경보를 검색하고 콘솔에 표시할 수 있습니다.
- sns 작업에 대한 권한이 있으면 콘솔에서 Amazon Simple Notification Service(Amazon SNS) 주제와 구독을 검색하고 콘솔에 표시할 수 있습니다.

## Amazon ElastiCache에 대한 AWS 관리형(미리 정의된) 정책

AWS는 AWS에서 생성하고 관리하는 독립형 IAM 정책을 제공하여 많은 일반 사용 사례를 처리합니다. 관리형 정책은 사용자가 필요한 권한을 조사할 필요가 없도록 일반 사용 사례에 필요한 권한을 부여합니다. 자세한 내용은 IAM 사용 설명서의 [AWS 관리형 정책](#) 단원을 참조하십시오.

계정의 사용자에게 연결할 수 있는 다음 AWS 관리형 정책은 ElastiCache에 대해 고유합니다.

- AmazonElastiCacheReadOnlyAccess - Amazon ElastiCache 리소스에 대한 읽기 전용 액세스 권한을 부여합니다.
- AmazonElastiCacheFullAccess - Amazon ElastiCache 리소스에 대한 모든 액세스 권한을 부여합니다.

### Note

IAM 콘솔에 로그인하고 이 콘솔에서 특정 정책을 검색하면 이러한 권한 정책을 검토할 수 있습니다.

Amazon ElastiCache API 작업에 대한 권한을 허용하는 고유의 사용자 지정 IAM 정책을 생성할 수도 있습니다. 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 사용자 지정 정책을 연결할 수 있습니다.

## 고객 관리형 정책에

기본 정책을 사용하지 않고 고객 관리형 정책을 사용할 경우 두 가지 중 하나가 가능한지 확인해야 합니다. `iam:createServiceLinkedRole`을 호출할 수 있는 권한이 있어야 합니다(자세한 내용은 [예제 5: 사용자에게 IAM CreateServiceLinkedRole API 호출 허용 \(p. 203\)](#) 단원을 참조하십시오). 또는 ElastiCache 서비스 연결 역할을 생성해야 합니다.

Amazon ElastiCache 콘솔을 사용하는 데 필요한 최소 권한과 함께 이 단원의 예제 정책을 사용할 경우 추가 권한이 부여됩니다. 예제는 AWS SDK 및 AWS CLI와도 관련이 있습니다. ElastiCache 콘솔을 사용하는 데 필요한 권한에 대한 자세한 내용은 [Amazon ElastiCache 콘솔 사용에 필요한 권한 \(p. 200\)](#) 단원을 참조하십시오.

IAM 사용자 및 그룹 설정 지침은 IAM 사용 설명서의 [첫 번째 IAM 사용자 및 관리자 그룹 생성](#)을 참조하십시오.

### Important

프로덕션에서 IAM 정책을 사용하기 전에 항상 철저히 테스트하십시오. 나타나는 일부 ElastiCache 작업을 ElastiCache 콘솔을 사용할 때 다른 작업이 해당 작업을 지원해야 할 수도 있습니다. 예를 들어, `elasticache:CreateCacheCluster`가 ElastiCache 캐시 클러스터를 생성하기 위한 권한을 부여합니다. 그러나 이 작업을 수행하기 위해 ElastiCache 콘솔에서 여러 `Describe` 및 `List` 작업을 사용하여 콘솔 목록을 채웁니다.

### 예

- [예제 1: 사용자에게 보안 그룹 생성 및 관리 허용 \(p. 201\)](#)
- [예제 2: 사용자에게 ElastiCache 리소스에 대한 읽기 전용 액세스 허용 \(p. 202\)](#)
- [예제 3: 사용자가 일반 ElastiCache 시스템 관리자 작업을 수행하도록 허용 \(p. 202\)](#)
- [예제 4: 사용자가 모든 ElastiCache API 작업에 액세스하도록 허용 \(p. 203\)](#)
- [예제 5: 사용자에게 IAM CreateServiceLinkedRole API 호출 허용 \(p. 203\)](#)

## 예제 1: 사용자에게 보안 그룹 생성 및 관리 허용

다음 정책은 보안 그룹의 특정한 ElastiCache 작업에 대한 권한을 부여합니다. 일반적으로 이 유형의 권한 정책을 시스템 관리자 그룹에 연결합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "SecGrpAllows",
 "Effect": "Allow",
 "Action": [
 "elasticache:CreateCacheSecurityGroup",
 "elasticache>DeleteCacheSecurityGroup",
 "elasticache:DescribeCacheSecurityGroup",
 "elasticache:AuthorizeCacheSecurityGroupIngress",
 "elasticache:RevokeCacheSecurityGroupIngress"],
 "Resource": "*"
]
}
```

### 예제 2: 사용자에게 ElastiCache 리소스에 대한 읽기 전용 액세스 허용

다음 정책은 사용자에게 리소스를 나열하도록 허용하는 ElastiCache 작업에 대한 권한을 부여합니다. 일반적으로 이 유형의 권한 정책을 관리자 그룹에 연결합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECUnrestricted",
 "Effect": "Allow",
 "Action": [
 "elasticache:Describe*",
 "elasticache:List*"],
 "Resource": "*"
]
}
```

### 예제 3: 사용자가 일반 ElastiCache 시스템 관리자 작업을 수행하도록 허용

일반 시스템 관리자 작업에는 캐시 클러스터, 파라미터 및 파라미터 그룹 수정이 포함됩니다. 시스템 관리자가 ElastiCache 이벤트에 대한 정보를 보고 싶어할 수도 있습니다. 다음 정책은 이러한 일반 시스템 관리자 작업을 위한 ElastiCache 작업을 수행할 권한을 사용자에게 부여합니다. 일반적으로 이 유형의 권한 정책을 시스템 관리자 그룹에 연결합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECAAllowSpecific",
 "Effect": "Allow",
 "Action": [
 "elasticache:ModifyCacheCluster",
 "elasticache:RebootCacheCluster",
 "elasticache:DescribeCacheClusters",
 "elasticache:DescribeEvents",
 "elasticache:ModifyCacheParameterGroup",
 "elasticache:DescribeCacheParameterGroups",
 "elasticache:DescribeCacheParameters",
 "elasticache:ResetCacheParameterGroup",
 "elasticache:DescribeEngineDefaultParameters"],
 "Resource": "*"
]
}
```

## 예제 4: 사용자가 모든 ElastiCache API 작업에 액세스하도록 허용

다음 정책은 사용자가 모든 ElastiCache 작업을 호출할 수 있도록 허용합니다. 관리자 사용자에게만 이 유형의 권한 정책을 부여하는 것이 좋습니다.

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Sid": "ECAAllowSpecific",
 "Effect": "Allow",
 "Action": [
 "elasticache:*"],
 "Resource": "*"
 }
]
```

## 예제 5: 사용자에게 IAM CreateServiceLinkedRole API 호출 허용

다음 정책은 사용자가 IAM CreateServiceLinkedRole API를 호출하도록 허용합니다. 변화하기 쉬운 ElastiCache 작업을 호출하는 사용자에게 이 유형의 권한 정책을 부여하는 것이 좋습니다.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CreateSLRAllows",
 "Effect": "Allow",
 "Action": [
 "iam:CreateServiceLinkedRole"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "iam:AWSServiceName": "elasticache.amazonaws.com"
 }
 }
 }
]
}
```

## Amazon ElastiCache에 서비스 연결 역할 사용

Amazon ElastiCache은 AWS Identity and Access Management(IAM) [서비스 연결 역할](#)을 사용합니다. 서비스 연결 역할은 AWS 서비스(예: Amazon ElastiCache, Amazon ElastiCache)에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 Amazon ElastiCache에서 사전 정의됩니다. 서비스에서 cluster를 대신 하여 AWS 서비스를 자동으로 호출하기 위해 필요한 모든 권한을 포함합니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할은 Amazon ElastiCache를 더 쉽게 설정할 수 있습니다. 이 역할은 AWS 계정에 이미 속해 있으나 Amazon ElastiCache 사용 사례에 연결되어 있으며 사전 정의된 권한을 가지고 있습니다. Amazon ElastiCache만 이 역할을 부여할 수 있고 이러한 역할만 사전 정의된 권한 정책을 사용할 수 있습니다. 먼저 역할의 관련 리소스를 삭제해야만 역할을 삭제할 수 있습니다. 이렇게 하면 Amazon ElastiCache 리소스에 대한 필수 액세스 권한을 부주의로 삭제할 수 없기 때문에 리소스가 보호됩니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 서비스 연결 역할 옆에 예가 있는 [IAM으로 작업하는 AWS 서비스](#)를 참조하십시오. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

목차

- [Amazon ElastiCache에 대한 서비스 연결 역할 권한 \(p. 204\)](#)
- [서비스 연결 역할 만들기\(IAM\) \(p. 205\)](#)
  - [서비스 연결 역할 만들기\(IAM 콘솔\) \(p. 205\)](#)
  - [서비스 연결 역할 만들기\(IAM CLI\) \(p. 205\)](#)
  - [서비스 연결 역할 만들기\(IAM API\) \(p. 206\)](#)
- [Amazon ElastiCache에 대한 서비스 연결 역할의 설명 편집 \(p. 206\)](#)
  - [서비스 연결 역할 설명 편집\(IAM 콘솔\) \(p. 206\)](#)
  - [서비스 연결 역할 설명 편집\(IAM CLI\) \(p. 206\)](#)
  - [서비스 연결 역할 설명 편집\(IAM API\) \(p. 207\)](#)
- [Amazon ElastiCache에 대한 서비스 연결 역할 삭제 \(p. 207\)](#)
  - [서비스 연결 역할 정리 \(p. 207\)](#)
  - [서비스 연결 역할 삭제\(IAM 콘솔\) \(p. 208\)](#)
  - [서비스 연결 역할 삭제\(IAM CLI\) \(p. 208\)](#)
  - [서비스 연결 역할 삭제\(IAM API\) \(p. 208\)](#)

## Amazon ElastiCache에 대한 서비스 연결 역할 권한

Amazon ElastiCache에서는 `AWSServiceRoleForElastiCache` – This policy allows ElastiCache to manage AWS resources on your behalf as necessary for managing your cache. 서비스 연결 역할을 사용합니다.

`AWSServiceRoleForElastiCache` 서비스 연결 역할 권한 정책은 Amazon ElastiCache가 지정된 리소스에서 다음 작업을 완료하도록 허용합니다.

```
Permission policy:
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ec2:AuthorizeSecurityGroupIngress",
 "ec2:CreateNetworkInterface",
 "ec2:CreateSecurityGroup",
 "ec2>DeleteNetworkInterface",
 "ec2>DeleteSecurityGroup",
 "ec2:DescribeAvailabilityZones",
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeSecurityGroups",
 "ec2:DescribeSubnets",
 "ec2:DescribeVpcs",
 "ec2:ModifyNetworkInterfaceAttribute",
 "ec2:RevokeSecurityGroupIngress",
],
 "Resource": "*"
 }
]
}
```

IAM 개체가 `AWSServiceRoleForElastiCache` 서비스 연결 역할을 생성하도록 허용하려면

IAM 개체에 대한 권한에 다음 정책 설명을 추가합니다.

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

IAM 개체가 AWSServiceRoleForElastiCache 서비스 연결 역할을 삭제하도록 허용하려면

IAM 개체에 대한 권한에 다음 정책 설명을 추가합니다.

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

또는 AWS 관리형 정책을 사용하여 Amazon ElastiCache에 대한 모든 액세스 권한을 제공할 수 있습니다.

## 서비스 연결 역할 만들기(IAM)

IAM 콘솔 CLI 또는 API를 사용하면 서비스 연결 역할을 만들 수 있습니다.

### 서비스 연결 역할 만들기(IAM 콘솔)

IAM 콘솔을 사용하여 서비스 연결 역할을 생성할 수 있습니다.

서비스 연결 역할을 만들려면(콘솔 사용)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다. 그런 다음 [Create new role]을 선택합니다.
3. AWS 서비스 연결 역할 단원을 확장한 후 새 서비스 연결 역할을 부여하려는 서비스를 선택합니다.
4. AWSServiceRoleForElastiCache 서비스 연결 역할 옆에 있는 선택을 선택합니다.
5. [Role name]에 서비스 연결 역할의 기본 이름에 추가할 접미사를 입력합니다. 이 접미사는 이 역할의 목적을 파악하는 데 도움이 됩니다. 역할 이름은 AWS 계정 내에서 고유해야 합니다. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **<service-linked-role-name>\_SAMPLE**과 **<service-linked-role-name>\_sample**, 두 가지로 지정된 역할을 만들 수는 없습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
6. (선택 사항) [Role description]에서 새로운 서비스 연결 역할에 대한 설명을 편집합니다.
7. 역할을 검토한 다음 [Create role]을 선택합니다.

### 서비스 연결 역할 만들기(IAM CLI)

AWS Command Line Interface에서 IAM 작업을 사용하여 서비스 연결 역할을 생성할 수 있습니다. 이 역할에는 서비스가 역할을 수임하는 데 필요한 신뢰 정책 및 인라인 정책이 포함될 수 있습니다.

서비스 연결 역할을 만들려면(CLI 사용)

다음 작업을 사용합니다.

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

## 서비스 연결 역할 만들기(IAM API)

IAM API를 사용하여 서비스 연결 역할을 생성할 수 있습니다. 이 역할에는 서비스가 역할을 수임하는 데 필요한 신뢰 정책 및 인라인 정책이 포함될 수 있습니다.

서비스 연결 역할을 만들려면(API 사용)

`CreateServiceLinkedRole` API 호출을 사용합니다. 요청 시 `elasticache.amazonaws.com` 서비스 이름을 지정합니다.

## Amazon ElastiCache에 대한 서비스 연결 역할의 설명 편집

Amazon ElastiCache에서는 `AWSServiceRoleForElastiCache` 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 개체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 그러나 IAM를 사용하여 역할의 설명을 편집할 수 있습니다.

### 서비스 연결 역할 설명 편집(IAM 콘솔)

IAM 콘솔을 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 편집하려면(콘솔 사용)

1. IAM 콘솔의 탐색 창에서 역할을 선택합니다.
2. 변경할 역할 이름을 선택합니다.
3. Role description(역할 설명)의 맨 오른쪽에서 편집을 선택합니다.
4. 상자에 새 설명을 입력하고 저장을 선택합니다.

### 서비스 연결 역할 설명 편집(IAM CLI)

AWS Command Line Interface의 IAM 작업을 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 변경하려면(CLI 사용)

1. (선택 사항) 역할에 대한 현재 설명을 보려면 IAM 작업 `get-role`에 대해 AWS CLI를 사용하십시오.

Example

```
$ aws iam get-role --role-name AWSServiceRoleForElastiCache
```

CLI 작업에서 역할을 참조하려면 ARN이 아니라 역할 이름을 사용해야 합니다. 예를 들어 어떤 역할의 ARN이 `arn:aws:iam::123456789012:role/myrole`인 경우 참조할 역할은 `myrole`입니다.

2. 서비스 연결 역할의 설명을 업데이트하려면 IAM 작업 `update-role-description`에 대해 AWS CLI를 사용하십시오.

Linux, macOS, or Unix의 경우:

```
$ aws iam update-role-description \
 --role-name AWSServiceRoleForElastiCache \
 --description "new description"
```

Windows의 경우:

```
$ aws iam update-role-description ^
```

```
--role-name AWSServiceRoleForElastiCache ^
--description "new description"
```

## 서비스 연결 역할 설명 편집(IAM API)

IAM API를 사용하여 서비스 연결 역할의 설명을 편집할 수 있습니다.

서비스 연결 역할의 설명을 변경하려면(API 사용)

1. (선택 사항) 역할의 현재 설명을 보려면 IAM API 작업 [GetRole](#)을 사용하십시오.

### Example

```
https://iam.amazonaws.com/
?Action=GetRole
&RoleName=AWSServiceRoleForElastiCache
&Version=2010-05-08
&AUTHPARAMS
```

2. 역할 설명을 업데이트하려면 IAM API 작업 [UpdateRoleDescription](#)을 사용하십시오.

### Example

```
https://iam.amazonaws.com/
?Action=UpdateRoleDescription
&RoleName=AWSServiceRoleForElastiCache
&Version=2010-05-08
&Description="New description"
```

## Amazon ElastiCache에 대한 서비스 연결 역할 삭제

서비스 연결 역할이 필요한 기능 또는 서비스가 더 이상 필요 없는 경우에는 해당 역할을 삭제할 것을 권합니다. 따라서 적극적으로 모니터링하거나 유지하지 않는 미사용 개체가 없도록 합니다. 단, 삭제 전에 서비스 연결 역할을 정리해야 합니다.

Amazon ElastiCache에서는 서비스 연결 역할을 자동으로 삭제하지 않습니다.

### 서비스 연결 역할 정리

IAM을 사용하여 서비스 연결 역할을 삭제하기 전에 먼저 역할에 해당 역할과 연결된 리소스(—클러스터—)가 없는지 확인합니다.

IAM 콘솔에서 서비스 연결 역할에 활성 세션이 있는지 확인하려면

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택합니다. 그런 다음 AWSServiceRoleForElastiCache 역할의 이름 (확인란 아님)을 선택합니다.
3. 선택한 역할의 [Summary] 페이지에서 [Access Advisor] 탭을 선택합니다.
4. [Access Advisor] 탭에서 서비스 연결 역할의 최근 활동을 검토합니다.

AWSServiceRoleForElastiCache가 필요한 Amazon ElastiCache 리소스를 삭제하는 방법(콘솔)

- 클러스터를 삭제하려면 다음을 참조하십시오.

- [AWS Management 콘솔 사용 \(p. 88\)](#)
- [AWS CLI 사용 \(p. 89\)](#)
- [ElastiCache API 사용 \(p. 89\)](#)

## 서비스 연결 역할 삭제(IAM 콘솔)

IAM 콘솔을 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할을 삭제하려면(콘솔)

1. AWS Management 콘솔에 로그인한 다음 <https://console.aws.amazon.com/iam/>에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 [Roles]를 선택합니다. 그런 다음 삭제할 역할의 이름이나 행이 아닌 이름 옆에 있는 확인란을 선택합니다.
3. 페이지 상단의 [Role actions]에서 [Delete role]을 선택합니다.
4. 확인 대화 상자가 나타나면 서비스 마지막 액세스 데이터를 검토합니다. 이 데이터는 선택한 각 역할이 AWS 서비스를 마지막으로 액세스한 일시를 보여 줍니다. 이를 통해 역할이 현재 활동 중인지 여부를 확인할 수 있습니다. 계속 진행하려면 [Yes, Delete]을 선택하여 삭제할 서비스 연결 역할을 제출합니다.
5. IAM 콘솔 알림을 보고 서비스 연결 역할 삭제 진행 상황을 모니터링합니다. IAM 서비스 연결 역할 삭제는 비동기이므로 삭제할 역할을 제출한 후에 삭제 작업이 성공하거나 실패할 수 있습니다. 작업에 실패할 경우 알림의 [View details] 또는 [View Resources]를 선택하면 삭제 실패 이유를 확인할 수 있습니다.

## 서비스 연결 역할 삭제(IAM CLI)

AWS Command Line Interface에서 IAM 작업을 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

서비스 연결 역할(CLI)을 삭제하려면

1. 삭제할 서비스 연결 역할의 이름을 모르는 경우 다음 명령을 입력합니다. 이 명령은 계정의 역할과 해당 Amazon 리소스 이름(ARN)을 나열합니다.

```
$ aws iam get-role --role-name role-name
```

CLI 작업에서 역할을 참조하려면 ARN이 아니라 역할 이름을 사용해야 합니다. 예를 들어 역할의 ARN이 `arn:aws:iam::123456789012:role/myrole`인 경우 해당 역할을 `myrole`으로 참조합니다.

2. 서비스 연결 역할이 사용되지 않거나 연결된 리소스가 없는 경우에는 서비스 연결 역할을 삭제할 수 없으므로 삭제 요청을 제출해야 합니다. 이러한 조건이 충족되지 않으면 요청이 거부될 수 있습니다. 삭제 작업 상태를 확인하려면 응답의 `deletion-task-id`를 캡처해야 합니다. 다음을 입력하여 서비스 연결 역할 삭제 요청을 제출합니다.

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. 다음을 입력하여 삭제 작업의 상태를 확인합니다.

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

삭제 작업은 NOT\_STARTED, IN\_PROGRESS, SUCCEEDED 또는 FAILED 상태일 수 있습니다. 삭제에 실패할 경우 문제를 해결할 수 있도록 실패 이유가 호출에 반환됩니다.

## 서비스 연결 역할 삭제(IAM API)

IAM API를 사용하여 서비스 연결 역할을 삭제할 수 있습니다.

## 서비스 연결 역할(API)을 삭제하려면

1. 서비스 연결 역할 삭제 요청을 제출하려면 `DeleteServiceLinkedRole`을 호출합니다. 요청에 역할 이름을 지정합니다.

서비스 연결 역할이 사용되지 않거나 연결된 리소스가 없는 경우에는 서비스 연결 역할을 삭제할 수 없으므로 삭제 요청을 제출해야 합니다. 이러한 조건이 충족되지 않으면 요청이 거부될 수 있습니다. 삭제 작업 상태를 확인하려면 응답의 `DeletionTaskId`를 캡처해야 합니다.

2. 삭제 상태를 확인하려면 `GetServiceLinkedRoleDeletionStatus`을 호출합니다. 요청에 `DeletionTaskId`를 지정합니다.

삭제 작업은 `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` 또는 `FAILED` 상태일 수 있습니다. 삭제에 실패할 경우 문제를 해결할 수 있도록 실패 이유가 호출에 반환됩니다.

## ElastiCache API 권한: 작업, 리소스 및 조건 참조

IAM 자격 증명에 연결할 액세스 제어 (p. 196) 및 쓰기 권한 정책(자격 증명 기반 정책)을 설정할 때 다음 표를 참조로 사용합니다. 표에는 각 Amazon ElastiCache API 작업과 이 작업을 수행할 수 있는 권한을 부여할 수 있는 작업이 나와 있습니다. 정책의 `Action` 필드에서 작업을 지정하고, 정책의 `Resource` 필드에서 리소스 값으로 와일드카드 문자 (\*)를 지정합니다.

ElastiCache 정책에서 AWS 차원 조건 키를 사용하여 조건을 표시할 수 있습니다. AWS 차원 키의 전체 목록은 IAM 사용 설명서의 [사용 가능한 조건 키](#) 단원을 참조하십시오.

### Note

작업을 지정하려면 `elasticache:` 접두사 다음에 API 작업 이름을 사용합니다(예: `elasticache:DescribeCacheClusters`). 모든 ElastiCache 작업에 와일드카드 문자(\*)를 리소스로 지정하십시오.

### Amazon ElastiCache API 및 작업에 대한 필수 권한

#### AddTagsToResource

작업: `elasticache:AddTagsToResource`

리소스: \*

#### AuthorizeCacheSecurityGroupIngress

작업: `elasticache:AuthorizeCacheSecurityGroupIngress`

리소스: \*

#### CreateCacheCluster

작업: `elasticache>CreateCacheCluster`

`s3:GetObject`

### Note

`SnapshotArns` 파라미터를 사용하는 경우 `SnapshotArns` 목록의 각 멤버에 `s3` ARN이 리소스인 자체 `s3:GetObject` 권한이 필요합니다.

리소스: \*

`arn:aws:s3:::my_bucket/snapshot1.rdb`

여기서 `my_bucket/snapshot1`은 캐시 클러스터를 생성할 S3 버킷 및 스냅샷입니다.

#### CreateCacheParameterGroup

작업: `elasticache:CreateCacheParameterGroup`

리소스: \*

#### CreateCacheSecurityGroup

작업: `elasticache:CreateCacheSecurityGroup`

리소스: \*

#### CreateCacheSubnetGroup

작업: `elasticache:CreateCacheSubnetGroup`

리소스: \*

#### DeleteCacheCluster

작업: `elasticache>DeleteCacheCluster`

리소스: \*

#### DeleteCacheParameterGroup

작업: `elasticache>DeleteCacheParameterGroup`

리소스: \*

#### DeleteCacheSecurityGroup

작업: `elasticache>DeleteCacheSecurityGroup`

리소스: \*

#### DeleteCacheSubnetGroup

작업: `elasticache>DeleteCacheSubnetGroup`

리소스: \*

#### DescribeCacheClusters

작업: `elasticache:DescribeCacheClusters`

리소스: \*

#### DescribeCacheEngineVersions

작업: `elasticache:DescribeCacheEngineVersions`

리소스: \*

#### DescribeCacheParameterGroups

작업: `elasticache:DescribeCacheParameterGroups`

리소스: \*

#### DescribeCacheParameters

작업: `elasticache:DescribeCacheParameters`

리소스: \*

#### DescribeCacheSecurityGroups

작업: `elasticache:DescribeCacheSecurityGroups`

리소스: \*

#### [DescribeCacheSubnetGroups](#)

작업: `elasticache:DescribeCacheSubnetGroups`

리소스: \*

#### [DescribeEngineDefaultParameters](#)

작업: `elasticache:DescribeEngineDefaultParameters`

리소스: \*

#### [DescribeEvents](#)

작업: `elasticache:DescribeEvents`

리소스: \*

#### [DescribeReservedCacheNodes](#)

작업: `elasticache:DescribeReservedCacheNodes`

리소스: \*

#### [DescribeReservedCacheNodesOfferings](#)

작업: `elasticache:DescribeReservedCacheNodesOfferings`

리소스: \*

#### [ListTagsForResource](#)

작업: `elasticache:ListTagsForResource`

리소스: \*

#### [ModifyCacheCluster](#)

작업: `elasticache:ModifyCacheCluster`

리소스: \*

#### [ModifyCacheParameterGroup](#)

작업: `elasticache:ModifyCacheParameterGroup`

리소스: \*

#### [ModifyCacheSubnetGroup](#)

작업: `elasticache:ModifyCacheSubnetGroup`

리소스: \*

#### [PurchaseReservedCacheNodesOffering](#)

작업: `elasticache:PurchaseReservedCacheNodesOffering`

리소스: \*

#### [RebootCacheCluster](#)

작업: `elasticache:RebootCacheCluster`

리소스: \*

#### [RemoveTagsFromResource](#)

작업: `elasticache:RemoveTagsFromResource`

리소스: \*

#### [ResetCacheParameterGroup](#)

작업: `elasticache:ResetCacheParameterGroup`

리소스: \*

#### [RevokeCacheSecurityGroupIngress](#)

작업: `elasticache:RevokeCacheSecurityGroupIngress`

리소스: \*

## Amazon ElastiCache 규정 준수 확인

타사 감사자는 여러 AWS 규정 준수 프로그램의 일환으로 Amazon ElastiCache의 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, FedRAMP, HIPAA 등이 포함됩니다.

규정 준수 프로그램 범위에 속하는 AWS 서비스의 목록은 [규정 준수 프로그램 제공 범위 내 AWS 서비스 페이지](#)에서 확인하십시오. 일반 정보는 [AWS 규정 준수 프로그램](#)을 참조하십시오.

AWS Artifact를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 [AWS 아티팩트의 보고서 다운로드](#)를 참조하십시오.

Amazon ElastiCache 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률 및 규정에 따라 결정됩니다. Amazon ElastiCache 사용 시 HIPAA, PCI 또는 Fedrmap와 같은 표준을 준수해야 하는 경우 다음과 같은 AWS 도움말 리소스를 활용하십시오.

- [보안 및 규정 준수 빠른 시작 안내서](#) – 이 배포 가이드에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 AWS 환경을 배포하기 위한 단계를 제공합니다.
- [HIPAA 보안 및 규정 준수 기술 백서 설계](#) – 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 만드는 방법을 설명합니다.
- [AWS 규정 준수 리소스](#) – 이 워크북 및 안내서는 귀사의 산업 및 위치에 적용될 수 있습니다.
- [AWS Config](#) – 이 AWS 서비스로 리소스 구성이 내부 관행, 업계 지침 및 규정을 준수하는 정도를 평가할 수 있습니다.
- [AWS Security Hub](#) – 이 AWS 서비스는 보안 업계 표준 및 모범 사례 준수 여부를 확인하는 데 도움이 되는 AWS 내 보안 상태에 대한 포괄적인 관점을 제공합니다.

## Amazon ElastiCache의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 리전에서는 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 [AWS 글로벌 인프라](#)를 참조하십시오.

AWS 글로벌 인프라 외에 Amazon ElastiCache도 데이터 복원성과 백업 요구 사항을 지원하는 다양한 기능을 제공합니다.

주제

- [장애 완화 \(p. 213\)](#)

## 장애 완화

Amazon ElastiCache 구현을 계획할 때는 장애가 애플리케이션 및 데이터에 미치는 영향을 최소화하도록 계획해야 합니다. 이 단원의 항목은 애플리케이션 및 데이터를 장애로부터 보호하기 위해 취할 수 있는 접근 방식을 다룹니다.

주제

- [Memcached 실행 시 장애 완화 \(p. 213\)](#)
- [권장 사항 \(p. 214\)](#)

## Memcached 실행 시 장애 완화

Memcached 엔진을 실행할 때 장애의 영향을 최소화하기 위한 다음과 같은 옵션이 있습니다. 장애 완화 계획에서 해결할 장애 유형에는 노드 장애와 가용 영역 장애의 두 유형이 있습니다.

### 노드 장애 완화

노드 장애의 영향을 완화하려면 캐시된 데이터를 더 많은 노드로 분산하십시오. Memcached에서는 복제를 지원하지 않으므로 노드 장애가 발생하면 항상 클러스터에서 일부 데이터가 손실됩니다.

Memcached 클러스터를 생성할 때 1 ~ 20개의 노드를 포함하여 생성하거나, 특별한 요청이 있는 경우 더 많은 노드를 포함하여 생성할 수 있습니다. 더 많은 노드로 데이터를 분할하면 노드에 장애가 발생할 경우 데이터 손실이 줄어듭니다. 예를 들어, 10개의 노드에 데이터를 분할하면 단일 노드는 캐시된 데이터의 약 10%를 저장합니다. 이 경우, 노드 장애로 인해 대체 노드가 생성되고 프로비저닝될 때 대체해야 하는 캐시의 약 10%가 손실됩니다. 동일한 데이터가 3개의 대형 노드에 캐시된 경우 노드의 장애가 발생하면 캐시된 데이터의 약 33%가 손실됩니다.

Memcached 클러스터에서 20개를 초과하는 노드가 필요할 경우 또는 AWS 리전에서 총 100개를 초과하는 노드가 필요할 경우 <https://aws.amazon.com/contact-us/elasticache-node-limit-request/>에서 ElastiCache 한도 증가 요청 양식을 작성하십시오.

Memcached 클러스터에서 노드 수를 지정하는 방법에 대한 자세한 내용은 [Memcached 클러스터 생성 \(콘솔\) \(p. 69\)](#) 단원을 참조하십시오.

### 가용 영역 장애 완화

가용 영역 장애의 영향을 완화하려면 가능한 한 많은 가용 영역에 노드를 배치합니다. 드물지만 AZ 장애가 발생할 경우 해당 AZ에 캐시된 데이터가 손실되며 다른 AZ에 캐시된 데이터는 손실되지 않습니다.

그러면 여러 개의 노드가 필요한 이유는 무엇입니까?

내 리전에 가용 영역이 3개만 있는 경우 AZ 장애 시 내 데이터의 약 1/3이 손실되므로 3개가 넘는 노드가 필요한 이유는 무엇입니까?

매우 좋은 질문입니다. 노드와 가용 영역이라는 두 개의 구분된 장애 유형을 완화하려고 시도하고 있습니다. 맞습니다. 가용 영역에 데이터가 분산되어 있으며 영역 중 하나에 장애가 발생하면 보유한 노드 수와 상관없이 해당 AZ에 캐시된 데이터만 손실됩니다. 그러나 노드 장애 시 더 많은 노드가 있으면 손실된 데이터의 비율이 감소됩니다.

클러스터에 포함할 노드 수를 결정하기 위한 "마법 공식"은 없습니다. 데이터 손실의 영향과 장애 가능성 및 비용을 비교하여 자체적인 결론을 내려야 합니다.

Memcached 클러스터에서 노드 수를 지정하는 방법에 대한 자세한 내용은 [Memcached 클러스터 생성 \(콘솔\) \(p. 69\)](#) 단원을 참조하십시오.

리전 및 가용 영역에 대한 자세한 내용은 [리전 및 가용 영역 선택 \(p. 30\)](#) 단원을 참조하십시오.

## 권장 사항

계획해야 할 장애의 유형에는 개별 노드 장애와 광범위한 가용 영역 장애의 두 유형이 있습니다. 가장 좋은 장애 완화 계획은 두 유형의 장애를 모두 해결하는 것입니다.

### 장애의 영향 최소화

노드 장애의 영향을 최소화하려면 구현 시 각 샤드에서 여러 노드를 사용하고 여러 가용 영역에 노드를 분산하는 것이 좋습니다.

Memcached를 실행하고 노드에 데이터를 분할할 때 더 많은 노드를 사용할수록 노드에 장애가 발생할 때 데이터 손실이 줄어듭니다.

### 가용 영역 장애의 영향 최소화

가용 영역 장애의 영향을 최소화하려면 가능한 한 여러 개의 서로 다른 가용 영역에서 노드를 시작하는 것이 좋습니다. AZ에 노드를 균등하게 분산하면, 드물지만 AZ 장애가 발생할 경우 영향을 최소화합니다.

## AWS ElastiCache의 인프라 보안

관리형 서비스인 AWS ElastiCache는 [Amazon Web Services: 보안 프로세스 개요](#) 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 보호됩니다.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 ElastiCache에 액세스합니다. 클라이언트가 TLS(전송 계층 보안) 1.0 이상을 지원해야 합니다. TLS 1.2 이상을 권장합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 PFS(전달 완전 보안, Perfect Forward Secrecy)가 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 보안 액세스 키를 사용하여 서명해야 합니다. 또는 [AWS Security Token Service\(AWS STS\)](#)를 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

# Amazon ElastiCache의 셀프 서비스 업데이트

Amazon ElastiCache는 Memcached 클러스터 및 노드 집합을 자동으로 모니터링하여 서비스 업데이트가 제공되면 이를 적용합니다. 이는 일반적으로 ElastiCache가 이 업데이트를 적용할 수 있도록 사전 정의된 유지 관리 기간을 설정해야 한다는 의미이며, 이 기간이 지나치게 엄격하여 비즈니스 흐름을 제한할 수도 있습니다. 셀프 서비스 업데이트를 통해 적용할 업데이트와 적용 시기를 제어할 수 있습니다. 선택한 Memcached 클러스터에 대한 업데이트 진행 상황을 실시간으로 모니터링할 수도 있습니다.

비즈니스 요구 사항에 따라 기존 노드 및 캐시 클러스터에 대한 업데이트를 중단할 수 있습니다. 서비스 업데이트가 완료되기 전까지 언제든지 적용되도록 새로운 클러스터 집합(일부 업데이트된 클러스터 포함)을 선택할 수 있습니다.

## 서비스 업데이트 관리

ElastiCache 서비스 업데이트는 정기적으로 릴리스됩니다. 서비스 업데이트에 대해 검증된 캐시 클러스터가 하나 이상 있는 경우 이메일, SNS, Personal Health Dashboard(PHD) 및 Amazon CloudWatch 이벤트를 통해 알림을 받게 됩니다. 업데이트는 ElastiCache 콘솔의 서비스 업데이트 페이지에도 표시됩니다.

그러나, 추천과 관계 없이 업데이트 적용 시기를 제어할 수 있습니다. Memcached 클러스터가 항상 현행 보안 패치로 최신 상태를 유지할 수 있도록 최소한 보안 유형의 업데이트는 모두 적용할 것을 강력하게 권장합니다.

자세한 내용은 [Amazon ElastiCache 유지 관리 도움말 페이지](#)를 참조하십시오.

다음 단원에서는 다음과 같은 옵션에 대해 상세히 알아봅니다.

주제

- [셀프 서비스 업데이트 적용 \(p. 215\)](#)
- [셀프 서비스 업데이트 중지 \(p. 220\)](#)

## 셀프 서비스 업데이트 적용

업데이트가 이용 가능 상태일 때부터 만료됨 상태가 될 때까지 Memcached 집합에 서비스 업데이트를 적용할 수 있습니다. 보안 유형의 서비스 업데이트는 축적됩니다. 즉, 만료되지 않은 상태로 아직 적용되지 않은 모든 업데이트가 최신 업데이트에 포함됩니다.

Note

날짜별 권장 적용의 기한이 지났더라도 사용 가능 상태인 서비스 업데이트만 적용할 수 있습니다.

Memcached 집합의 검토와 적용 가능한 Memcached 클러스터에 서비스별 업데이트의 적용에 관한 자세한 정보는 [Memcached용 콘솔을 사용한 서비스 업데이트 적용 \(p. 216\)](#) 단원을 참조하십시오.

집합 내 하나 이상의 Memcached 클러스터에 새로운 서비스 업데이트가 제공될 경우, ElastiCache 콘솔, API 또는 AWS CLI를 사용하여 이 업데이트를 적용할 수 있습니다. 다음 단원에서는 업데이트 적용에 사용할 수 있는 옵션에 대해 설명합니다.

주제

- 콘솔을 사용한 서비스 업데이트 적용 (p. 216)
- AWS CLI를 사용한 서비스 업데이트 적용 (p. 220)

## 콘솔을 사용한 서비스 업데이트 적용

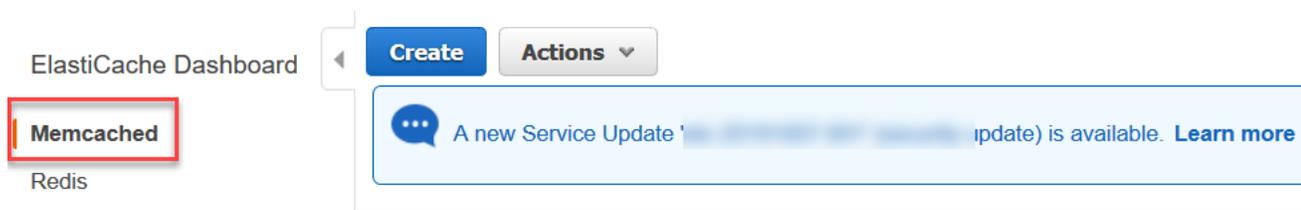
다음의 콘솔 옵션 중 하나를 사용하여 서비스 업데이트를 적용할 수 있습니다. ElastiCache는 서로 다른 두 가지 관점을 통해 업데이트의 적용 방법과 시기를 판단할 수 있도록 지원합니다.

주제

- Memcached용 콘솔을 사용한 서비스 업데이트 적용 (p. 216)
- 서비스 업데이트 목록을 사용한 서비스 업데이트 적용 (p. 218)

## Memcached용 콘솔을 사용한 서비스 업데이트 적용

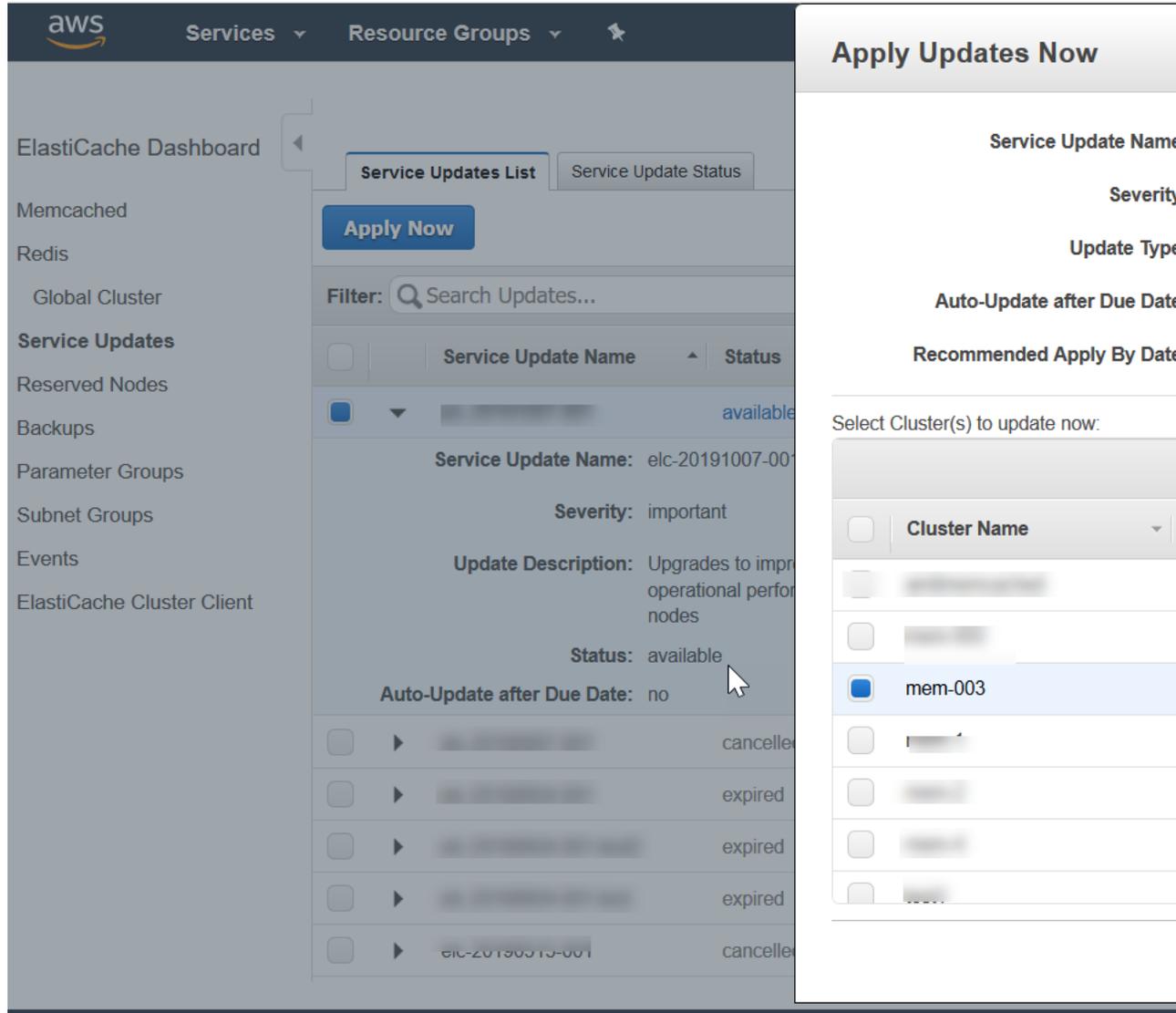
개별 Memcached 클러스터의 업데이트 상태를 검토하려면 이 옵션을 선택한 다음, 해당 서비스 업데이트에 대해 적용, 보기 또는 중지를 선택하십시오. 서비스 업데이트가 제공되면 콘솔의 Memcached 페이지 상단에 다음과 같이 배너가 표시됩니다.



- 지금 적용을 선택하면 다음과 같이 해당 워크플로 내의 적용 가능한 모든 클러스터 또는 하위 집합에 서비스 업데이트를 적용하기로 선택할 수 있습니다.

### Note

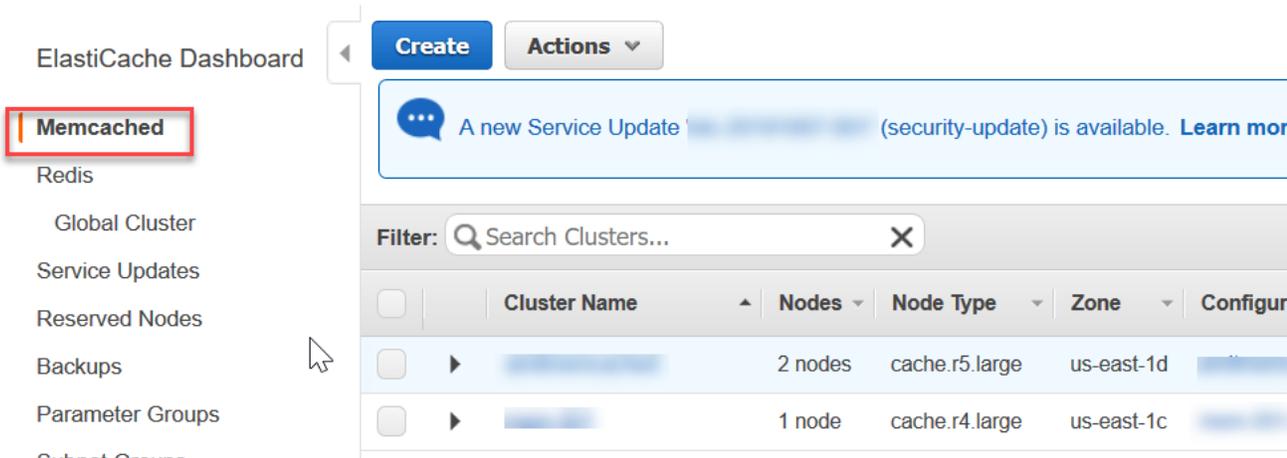
무시를 선택하면 해당 콘솔 세션 중에는 콘솔에 배너가 더 이상 표시되지 않습니다. 그러나, 세션을 새로 고침한 후에는 배너가 다시 나타납니다.



지금 업데이트 적용 페이지에 대한 다음 사항에 유의하십시오.

- Auto-Update after Due Date(기한 이후 자동 업데이트): 셀프 서비스 업데이트 만료 전에 적용하지 않을 경우, 업데이트되지 않은 모든 클러스터 또는 개별 노드가 다음 번 누적 업데이트 출시 전까지 규정 미준 수 상태로 남아 있게 됩니다 ElastiCache는 사용자를 대신해 서비스 업데이트를 자동 적용하지 않습니다.
- Memcached 클러스터의 Nodes Updated(업데이트된 노드) 비율과 Estimated Update Time(예상 업데이트 소요 시간)을 통해 유지 관리 일정을 계획할 수 있습니다. 서비스 업데이트가 비즈니스 흐름의 예상 소요 시간 제약을 초과한 경우, 업데이트를 중단하고 나중에 다시 적용할 수 있습니다. 자세한 내용은 [셀프 서비스 업데이트 중지 \(p. 220\)](#) 단원을 참조하십시오.
- 서비스 업데이트를 가능한 모든 Memcached 클러스터에 적용하기로 할 경우, 확인을 선택하십시오. 이를 선택하면, 서비스 업데이트의 상태를 모니터링할 수 있는 서비스 업데이트 페이지를 확인할 수 있습니다.
- 취소를 선택하면 다음과 같은 다른 옵션을 알아 볼 수 있습니다.

Memcached 클러스터를 개별적으로 검사하여 업데이트 상태를 판단할 수 있습니다. 다음을 통해 이용 가능한 서비스 업데이트와 관련된 클러스터의 규정 준수 상태를 알 수 있습니다.



업데이트 상태는 다음 중 하나를 표시합니다.

- 업데이트 가능: 이 클러스터에 업데이트를 적용할 수 있습니다.
- 진행 중: 업데이트가 이 클러스터에 적용되고 있으므로 비즈니스 흐름에 이 클러스터를 사용할 수 없는 상태입니다.
- 중지 중: 진행 중인 업데이트가 완료 전에 중지되었습니다.
- 중지됨: 업데이트가 종료되었습니다.

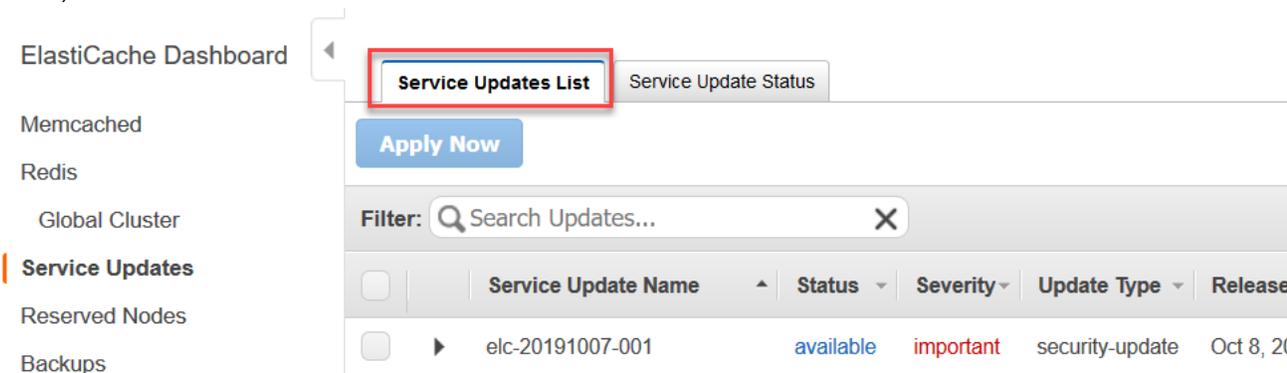
#### Note

Memcached 클러스터에 대해 진행 중인 업데이트를 중지하면 일부 노드는 업데이트되고 다른 노드는 업데이트되지 않습니다. 중지 중 프로세스는 이미 업데이트된 노드의 변경 사항을 되돌리지 않습니다. 아직 사용 가능 상태인 업데이트는 편의에 따라 만료됨 상태가 되기 전까지 노드에 다시 적용할 수 있습니다.

- 최신 상태: 업데이트가 적용되었으며 클러스터가 최신 상태입니다.

## 서비스 업데이트 목록을 사용한 서비스 업데이트 적용

다른 관련 정보와 함께 개별 서비스 업데이트 목록과 상태를 검토하려면 Service Updates List(서비스 업데이트 목록) 탭을 선택하십시오.



Service Updates List(서비스 업데이트 목록)을 확인할 때는 다음 사항을 참고하십시오.

- 서비스 업데이트 이름: 서비스 업데이트의 고유 식별자입니다.
- 상태: 업데이트의 상태로, 다음 중 하나에 해당합니다.

- 사용 가능: 업데이트를 필요한 Memcached 클러스터에 사용할 수 있습니다.
- 완료: 업데이트가 적용되었으며 모든 Memcached 클러스터가 최신 상태입니다.
- 취소됨: 업데이트가 취소되었으며 더 이상 필요하지 않습니다.
- 만료됨: 업데이트를 더 이상 적용할 수 없습니다.
- 심각도: 업데이트 적용의 우선 순위를 나타냅니다.
  - 심각: 즉시 적용하는 것이 좋습니다(14일 이내).
  - 중요: 비즈니스 흐름 상 가능한 한 빨리 적용하는 것이 좋습니다(30일 이내).
  - 중간: 비즈니스 흐름 상 가능한 한 빨리 적용하는 것이 좋습니다(60일 이내)
  - 낮음: 비즈니스 흐름 상 가능한 한 빨리 적용하는 것이 좋습니다(90일 이내)
- 업데이트 유형: 이 버전에는 보안 업데이트만 지원됩니다.
- 릴리스 날짜: 업데이트가 릴리스되어 Redis 집합에 적용할 수 있게 된 날짜입니다.
- Recommended Apply By Date(날짜별 권장 적용): 업데이트를 적용해야 하는 ElastiCache 지침 일자입니다.

개별 업데이트를 선택하면 다음과 같이 추가적인 세부 정보가 제공됩니다.

- 업데이트 설명: 서비스 업데이트에 관한 세부 정보를 제공합니다.
- 업데이트 만료 일자: 서비스 업데이트가 만료되어 더 이상 제공되지 않는 날짜를 의미합니다. 만료 일자 이전에 적용되지 않은 모든 업데이트는 다음 업데이트로 누적됩니다.

해당하는 Memcached 클러스터와 관련된 개별 서비스 업데이트 목록을 검토하려면 Service Update Status(서비스 업데이트 상태) 탭을 선택하십시오.

The screenshot shows the Amazon ElastiCache console interface. On the left is a navigation menu with 'Service Updates' selected. The main content area shows the 'Service Update Status' tab, which is highlighted with a red box. Below the tab is a search filter and a table of updates. The table has columns for 'Service Update Name', 'Cluster Name', 'Engine', 'Nodes Updated', and 'Update Status'. Three rows of update data are visible, showing different update names, cluster names, engines, and the number of nodes updated (e.g., 1/2 nodes, 3/3 nodes, 9/9 nodes).

Service Updates Status(서비스 업데이트 상태) 목록을 확인할 때는 다음 사항을 참고하십시오.

- 서비스 업데이트 이름: 서비스 업데이트에 관한 상세 정보를 제공합니다.
- 클러스터 이름: 업데이트 대상인 Memcached 클러스터의 목록입니다.
- 업데이트된 노드: 업데이트되었거나 특정 서비스 업데이트를 여전히 사용할 수 있는 특정 클러스터 내 개별 노드의 비율입니다.
- 업데이트 심각도: 업데이트 적용의 우선 순위를 나타냅니다.
  - 심각: 즉시 적용하는 것이 좋습니다(14일 이내).
  - 중요: 비즈니스 흐름 상 가능한 한 빨리 적용하는 것이 좋습니다(30일 이내).
  - 중간: 비즈니스 흐름 상 가능한 한 빨리 적용하는 것이 좋습니다(60일 이내)
  - 낮음: 비즈니스 흐름 상 가능한 한 빨리 적용하는 것이 좋습니다(90일 이내)

- 업데이트 유형: 이 버전에는 보안 업데이트만 지원됩니다.
- Service Update Status(서비스 업데이트 상태): 업데이트의 상태로, 다음 중 하나에 해당합니다.
  - 사용 가능: 업데이트를 필요한 Memcached 클러스터에 사용할 수 있습니다.
  - 완료: 업데이트가 적용되었으며 모든 Memcached 클러스터가 업데이트되었습니다.
  - 취소됨: 업데이트가 취소되었으며 더 이상 필요하지 않습니다.
  - 만료됨: 업데이트를 더 이상 적용할 수 없습니다.
- Service Update SLA Met(서비스 업데이트 SLA 충족됨): 클러스터가 규정을 준수하는지 여부를 반영합니다.
  - 예: 제공되는 모든 업데이트가 이 클러스터와 사용 가능한 노드에 적용되었습니다.
  - 아니요: 서비스 업데이트가 하나 이상의 노드에 성공적으로 적용되었을 수 있으나 클러스터 내의 다른 노드는 여전히 사용 가능 상태입니다. 이는 보통 서비스 업데이트가 적용되었다가 중지된 경우에 나타납니다.

#### Note

클러스터에 대한 서비스 업데이트 진행을 중지할 경우에도 이미 업데이트된 노드는 완료 상태가 됩니다. 진행 중 또는 중지 중 상태인 노드는 중지됨 상태가 되고 Service Update SLA Met(서비스 업데이트 SLA 충족됨) 상태가 아니므로 변경됩니다.

- Cluster Status Modified Date(클러스터 상태 수정 날짜): 서비스 업데이트로 인해 클러스터가 수정된 최근 날짜입니다.

#### Note

Show Previous Updates(이전 업데이트 보기) 확인란을 선택하면 더 이상 제공되지 않는 이전 업데이트의 목록이 표시됩니다.

## AWS CLI를 사용한 서비스 업데이트 적용

서비스 업데이트가 제공된다는 알림을 받은 후 AWS CLI를 사용하여 해당 업데이트를 검사하고 적용할 수 있습니다.

- 제공되는 서비스 업데이트의 설명을 검색하려면:

```
aws elasticache describe-service-updates --service-update-status available
```

자세한 정보는 [DescribeServiceUpdates](#)를 참조하십시오.

- not-applied 또는 stopped 상태인 업데이트 작업을 검토하려면:

```
aws elasticache describe-update-actions --service-update-name sample-service-update --update-action-status not-applied stopped
```

자세한 내용은 [DescribeUpdateActions](#)를 참조하십시오.

- 복제본 그룹 목록에 서비스 업데이트를 적용하려면:

```
aws elasticache batch-apply-update-action --service-update-name sample-service-update --cache-cluster-ids my-cachecluster-1 my-cache-cluster-2
```

자세한 정보는 [BatchApplyUpdateAction](#)을 참조하십시오.

## 셀프 서비스 업데이트 중지

업데이트를 진행 중인 Redis 클러스터에 예상치 못한 서지가 발생하거나, 업데이트에 소요되는 시간이 지나치게 길어 비즈니스 흐름 피크 타임에 방해가 되는 경우 업데이트를 중지할 수 있습니다.

**중지 중** 작업은 이들 노드와 아직 업데이트되지 않은 노드에 대한 모든 업데이트를 즉시 중지합니다. 진행 중 상태인 노드는 완료될 때까지 계속됩니다. 그러나, 업데이트 사용 가능 상태인 동일한 클러스터 내의 다른 노드에 대한 업데이트는 중단되며 **중지 중** 상태로 변경됩니다.

중지 중 워크플로가 완료되면 **중지 중** 상태인 노드는 중지됨 상태가 됩니다. 업데이트의 워크플로에 따라 일부 클러스터의 노드는 업데이트되지 않습니다. 다른 클러스터에는 업데이트된 노드와 여전히 업데이트 사용 가능 상태인 노드가 포함될 수 있습니다.

비즈니스 흐름 상 가능할 때 업데이트 프로세스를 완료할 수 있습니다. 이 경우, 업데이트를 완료할 해당 클러스터를 선택한 다음 **지금 적용**을 선택하십시오. 자세한 내용은 [Memcached용 콘솔을 사용한 서비스 업데이트 적용 \(p. 216\)](#) 단원을 참조하십시오.

주제

- [콘솔을 사용한 서비스 업데이트 중지 \(p. 221\)](#)
- [AWS CLI를 사용한 서비스 업데이트 중지 \(p. 222\)](#)

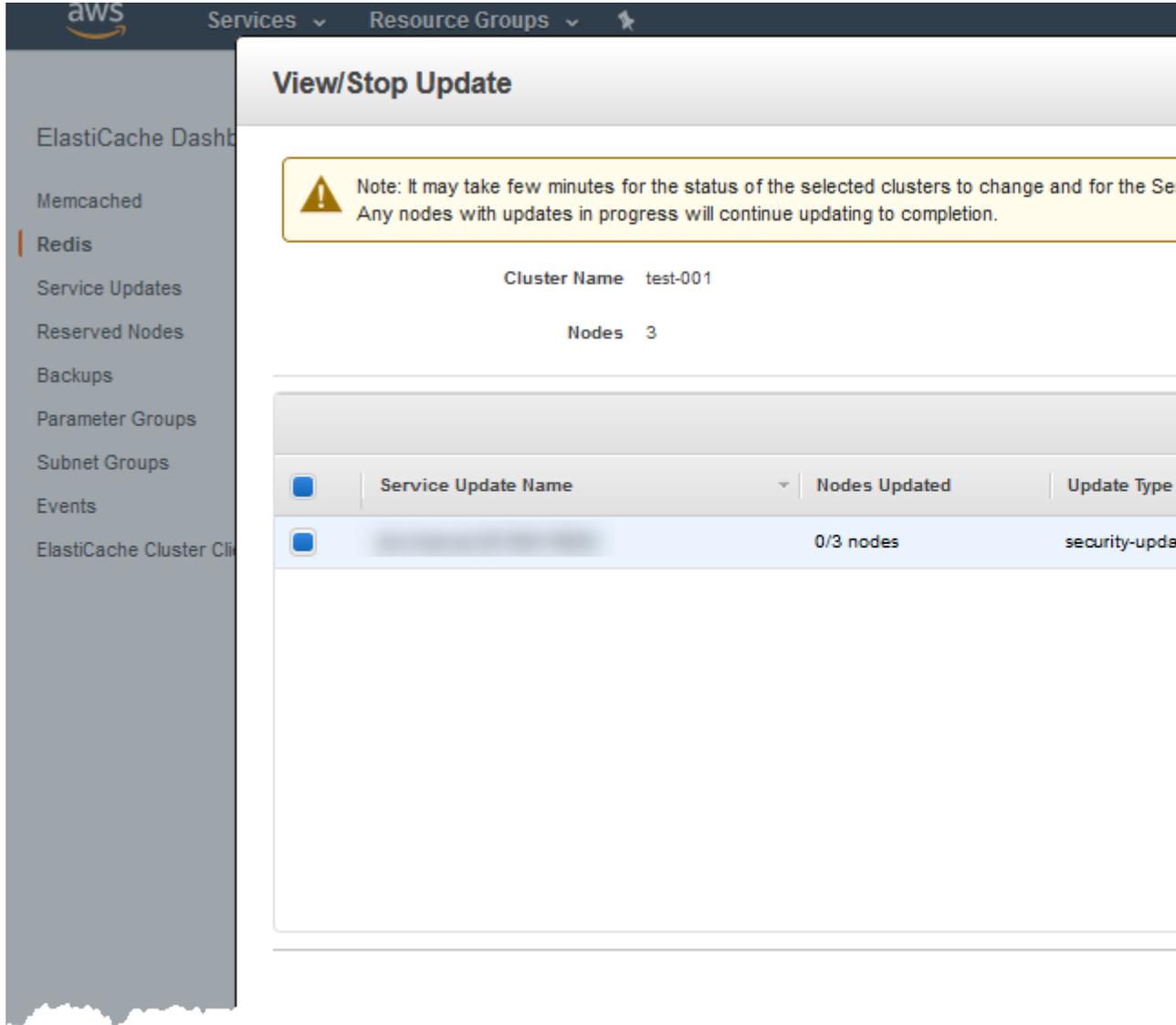
## 콘솔을 사용한 서비스 업데이트 중지

콘솔을 사용해 서비스 업데이트를 중지할 수 있습니다. 다음에서는 이 작업을 수행하는 방법을 보여 줍니다.

- 선택한 Memcached 클러스터에 서비스 업데이트가 진행된 후 ElastiCache 콘솔의 대시보드 상단에 **View/Stop Update**(업데이트 보기/중지) 탭이 나타납니다.



- 업데이트를 중지하려면 **Stop Update**(업데이트 중지)를 선택합니다.



- 업데이트를 중지할 경우 Memcached 클러스터를 선택하고 상태를 확인하십시오. 중지 중 상태로 바뀌었다가 최종적으로 중지됨 상태가 됩니다.

## AWS CLI를 사용한 서비스 업데이트 중지

AWS CLI를 사용해 서비스 업데이트를 중지할 수 있습니다. 다음 코드 예제에서는 이를 수행하는 방법을 보여 줍니다.

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-group-ids my-cache-group-1 my-cache-group-2
```

자세한 정보는 [BatchStopUpdateAction](#)을 참조하십시오.

## 참조

이 단원의 본 주제에서는 Amazon ElastiCache API 및 AWS CLI의 ElastiCache 섹션 관련 작업을 다룹니다. 또한 여기에는 일반적인 오류 메시지와 서비스 알림에 관한 설명이 포함되어 있습니다.

- [ElastiCache API 사용 \(p. 223\)](#)
- [ElastiCache API 참조](#)
- [AWS CLI 참조의 ElastiCache 단원](#)
- [Amazon ElastiCache 오류 메시지 \(p. 233\)](#)
- [알림 \(p. 234\)](#)

## ElastiCache API 사용

이 단원에서는 ElastiCache 작업 사용 방법 및 구현에 관해 작업 중심으로 설명합니다. 이러한 작업 설명 전체를 보려면 [Amazon ElastiCache API Reference](#) 단원을 참조하십시오.

주제

- [Query API 사용 \(p. 223\)](#)
- [사용 가능한 라이브러리 \(p. 225\)](#)
- [애플리케이션 문제 해결 \(p. 226\)](#)
- [AWS CloudTrail을 사용하여 Amazon ElastiCache API 호출 로깅 \(p. 226\)](#)

## Query API 사용

### 쿼리 매개 변수

HTTP 쿼리 기반 요청은 GET 또는 POST와 같은 HTTP 동사와 Action 쿼리 매개 변수를 사용하는 HTTP 요청입니다.

각 쿼리 요청은 인증 및 작업을 처리할 수 있도록 일부 공통 파라미터를 포함해야 합니다.

일부 작업은 파라미터의 목록을 허용합니다. 이러한 목록은 `param.n` 표기법을 사용하여 지정됩니다. `n`의 값은 1부터 시작하는 정수입니다.

### 쿼리 요청 인증

HTTPS를 통해서만 쿼리 요청을 보낼 수 있으며 모든 쿼리 요청에는 서명이 포함되어야 합니다. 이 단원에서는 서명을 작성하는 방법을 설명합니다. 아래 절차에 설명된 방법은 서명 버전 4라고 합니다.

다음은 AWS에 대한 요청을 인증하는 데 사용되는 기본 단계입니다. 이 경우 사용자가 AWS에 등록되어 있으며 액세스 키 ID 및 보안 액세스 키를 가지고 있다고 가정합니다.

쿼리 인증 절차

1. 발신자가 AWS에 대한 요청을 구성합니다.
2. 발신자가 이 항목의 다음 단원에 정의된 방법으로 SHA-1 해시 기능을 사용하는 HMAC(Hash-based Message Authentication Code)에 대한 키 해싱인 요청 서명을 계산합니다.
3. 요청의 발신자가 요청 데이터, 서명 및 액세스 키 ID(사용된 보안 액세스 키의 키 식별자)를 AWS로 보냅니다.

4. AWS는 액세스 키 ID를 사용하여 보안 액세스 키를 찾습니다.
5. AWS는 요청의 서명 계산에 사용된 동일한 알고리즘을 사용하여 요청 데이터 및 보안 액세스 키에서 서명을 생성합니다.
6. 서명이 일치하는 경우 요청이 인증되는 것으로 간주됩니다. 서명이 일치하지 않는 경우 요청이 삭제되고 AWS는 오류를 반환합니다.

#### Note

Timestamp 매개 변수가 요청에 포함된 경우 요청에 대해 계산된 서명은 그 매개 변수 값보다 15분 후에 만료됩니다.

Expires 매개 변수가 요청에 포함된 경우 그 서명은 Expires 매개 변수에 의해 지정된 시간에 만료됩니다.

#### 요청 서명을 계산하려면

1. 정규화된 쿼리 문자열을 만듭니다. 이 절차의 뒷부분에서 필요합니다.
  - a. UTF-8 쿼리 문자열 구성 요소를 매개 변수 이름의 일반 바이트 순서로 정렬합니다. 이 매개 변수는 GET URI 또는 POST 요청 본문(Content-Type이 application/x-www-form-urlencoded일 경우)의 내용이 사용될 수 있습니다.
  - b. 다음 규칙에 따라 매개 변수 이름과 값을 URL 인코딩합니다.
    - i. RFC 3986에 정의된 예약되지 않은 모든 문자는 URL 인코딩하지 않습니다. 이러한 예약되지 않은 문자는 A~Z, a~z, 0~9, 하이픈(-), 밑줄(\_), 마침표(.) 및 물결표(~)입니다.
    - ii. %XY와 같이 모든 기타 문자를 퍼센트 인코딩합니다(여기서 X 및 Y는 16진 문자 0~9 및 대문자 A~F).
    - iii. 확장된 UTF-8 문자는 %XY%ZA... 형식으로 퍼센트 인코딩합니다.
    - iv. 공백 문자는 %20(일반 인코딩 구조인 +가 아님)으로 퍼센트 인코딩합니다.
  - c. 매개 변수 값이 비어있는 경우에도 인코딩된 매개 변수 이름을 인코딩된 매개 변수 값과 등호(=) (ASCII 문자 61)로 구별합니다.
  - d. 앰퍼샌드(&)(ASCII 코드 38)로 이름-값 쌍을 구별합니다.
2. 다음의 의사(pseudo) 문법("\n"은 ASCII 줄 바꿈을 나타냄)에 따라 서명할 문자열을 만듭니다.

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

HTTPRequestURI 구성 요소는 URI의 HTTP 절대 경로 구성 요소이고 쿼리 문자열은 포함하지는 않습니다. HTTPRequestURI가 비어있는 경우 슬래시(/)를 사용합니다.

3. 사용자의 보안 액세스 키를 키로, SHA256 또는 SHA1을 해시 알고리즘으로 하여 방금 만든 문자열로 RFC 2104 호환 HMAC를 계산합니다.

자세한 내용은 <https://www.ietf.org/rfc/rfc2104.txt>를 참조하십시오.

4. 결과 값을 base64로 변환합니다.
5. 요청에서 signature 매개 변수 값을 값으로 포함합니다.

예를 들어, 다음은 샘플 요청입니다(줄 바꿈이 명확성을 위해 추가됨).

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterIdentifier=myCacheCluster
```

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
사용 가능한 라이브러리

```
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-12-01
```

이전 쿼리 문자열의 경우 다음 문자열을 통해 HMAC 서명을 계산합니다.

```
GET\n
elasticache.amazonaws.com\n
Action=DescribeCacheClusters
&CacheClusterIdentifier=myCacheCluster
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-12-01
&X-Amz-Algorithm=AWS4-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:
```

이 결과는 다음의 서명된 요청입니다.

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterIdentifier=myCacheCluster
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-12-01
&X-Amz-Algorithm=AWS4-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
```

서명 프로세스 및 요청 서명 계산에 대한 자세한 내용은 [서명 버전 4 서명 프로세스](#) 항목과 그 하위 항목을 참조하십시오.

## 사용 가능한 라이브러리

AWS는 Query API 대신 언어별 API를 사용하여 애플리케이션을 구축하려는 소프트웨어 개발자를 위한 소프트웨어 개발 키트(SDK)를 제공합니다. 이러한 SDK는 보다 쉽게 시작하도록 요청 인증, 요청 재시도 및 오류 처리 같은 기본 기능(API에는 포함되지 않음)을 제공합니다. SDK 및 추가 리소스는 다음 프로그래밍 언어에 대해 제공됩니다.

- [Java](#)
- [Windows 및 .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

다른 언어에 대한 자세한 내용은 [샘플 코드 및 라이브러리](#)를 참조하십시오.

## 애플리케이션 문제 해결

ElastiCache는 ElastiCache API와 상호 작용하는 동안 발생하는 문제를 해결할 때 도움이 되도록 구체적이고 서술적인 오류를 제공합니다.

### 오류 검색

일반적으로 사용자는 시간을 소비하여 결과를 처리하기 전에 애플리케이션이 먼저 해당 요청으로 오류가 발생하는지 여부를 확인하려고 합니다. 오류 발생 여부를 확인하는 가장 쉬운 방법은 ElastiCache API의 응답에서 `Error` 노드를 찾는 것입니다.

XPath 구문은 `Error` 노드의 발생뿐만 아니라 오류 코드 및 메시지를 쉽게 검색할 수 있는 간단한 방법을 제공합니다. 다음 코드 조각에서는 요청 중에 오류가 발생했는지 여부를 파악하기 위해 Perl 및 XML::XPath 모듈을 사용합니다. 오류가 발생되면 코드는 응답에 첫 번째 오류 코드와 메시지를 인쇄합니다.

```
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"; }
```

### 문제 해결 팁

다음 절차를 통해 ElastiCache API의 문제를 진단하고 해결하는 것이 좋습니다.

- ElastiCache가 올바르게 실행되는지 확인합니다.

이렇게 하려면, 브라우저 창을 열고 ElastiCache 서비스(<https://elasticache.amazonaws.com> 등)에 쿼리 요청을 제출하면 됩니다. `MissingAuthenticationTokenException` 또는 500 내부 서버 오류가 발생하는 경우 서비스가 사용 가능하고 요청에 응답하는지 확인할 수 있습니다.

- 요청 구조 확인.

각 ElastiCache 작업에 대한 참조 페이지는 ElastiCache API 참조에 있습니다. 파라미터를 올바르게 사용하고 있는지 여부를 다시 확인합니다. 어떤 문제가 발생할 수 있을 지에 대해 미리 알아보려면 샘플 요청이나 사용자 시나리오를 살펴보고 이러한 샘플이 유사한 작업을 하고 있는지 확인하십시오.

- 포럼 확인.

ElastiCache에는 그 과정에서 다른 사람들이 경험한 문제에 대한 해결책을 검색할 수 있는 토론 포럼이 있습니다. 포럼을 보려면 다음 사이트를 참조하십시오.

<https://forums.aws.amazon.com/> .

## AWS CloudTrail을 사용하여 Amazon ElastiCache API 호출 로깅

Amazon ElastiCache는 Amazon ElastiCache의 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 레코드를 제공하는 서비스인 AWS CloudTrail과 통합됩니다. CloudTrail은 Amazon ElastiCache 콘솔의 호출 및 Amazon ElastiCache API 코드 호출 등 Amazon ElastiCache에 대한 모든 API 작업 호출을 이벤트로 캡처합니다. 추적을 생성하면 Amazon ElastiCache에 대한 이벤트를 비롯하여 CloudTrail 이벤트를 Amazon S3 버킷으로 지속적으로 배포할 수 있습니다. 추적을 구성하지 않은 경우 Event history(이벤트 기록)에서 CloudTrail 콘솔의 최신 이벤트를 볼 수도 있습니다. CloudTrail에서 수집하는 정보를 사용하여 Amazon ElastiCache에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail User Guide](#)를 참조하십시오.

## CloudTrail의 Amazon ElastiCache 정보

CloudTrail은 계정 생성 시 AWS 계정에서 활성화됩니다. Amazon ElastiCache에서 활동이 수행되면 해당 활동은 Event history(이벤트 기록)에서 다른 AWS 서비스 이벤트와 함께 CloudTrail 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 [CloudTrail 이벤트 기록에서 이벤트 보기](#)를 참조하십시오.

Amazon ElastiCache에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려는 경우 추적을 생성합니다. 추적은 CloudTrail이 Amazon S3 버킷으로 로그 파일을 전송할 수 있도록 합니다. 콘솔에서 추적을 생성하면 기본적으로 모든 리전에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또는 CloudTrail 로그에서 수집된 이벤트 데이터를 추가 분석 및 처리하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기](#)

모든 Amazon ElastiCache 작업이 CloudTrail에서 로깅되고 [ElastiCache API Reference](#)에 문서화됩니다. 예를 들어, `CreateCacheCluster`, `DescribeCacheCluster` 및 `ModifyCacheCluster` 작업을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 자격 증명 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 IAM 사용자 자격 증명으로 했는지 여부
- 역할 또는 연합된 사용자에 대한 임시 보안 자격 증명을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail userIdentity 요소](#)를 참조하십시오.

## Amazon ElastiCache 로그 파일 항목 이해

추적은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 제공할 수 있도록 해 주는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 어떤 소스로부터의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 주문 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

다음은 `CreateCacheCluster` 작업을 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다.

```
{
 "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",
```

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
AWS CloudTrail을 사용하여  
Amazon ElastiCache API 호출 로깅

```

"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.01",
"userAgent": "Amazon 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"
}

```

다음은 DescribeCacheCluster 작업을 보여 주는 CloudTrail 로그 항목이 나타낸 예제입니다. 모든 Amazon ElastiCache Describe 호출(Describe\*)의 경우 ResponseElements 단원이 제거되며 null으로 표시됩니다.

```

{
 "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": "Amazon CLI/ElastiCache 1.10 API 2014-12-01",
 "requestParameters": {
 "showCacheNodeInfo": false,
 "maxRecords": 100
 }
}

```

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서  
AWS CloudTrail을 사용하여  
Amazon ElastiCache API 호출 로깅

```
},
"responseElements":null,
"requestID":"1f0b5031-3548-11e4-9376-c1d979ba565a",
"eventID":"a58572a8-e81b-4100-8e00-1797ed19d172"
}
```

다음 예제는 ModifyCacheCluster 작업을 기록하는 CloudTrail 로그 항목을 보여줍니다.

```
{
 "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":"Amazon CLI/ElastiCache 1.10 API 2014-12-01",
 "requestParameters":{
 "applyImmediately":true,
 "numCacheNodes":3,
 "cacheClusterId":"test-memcached"
 },
 "responseElements":{
 "engine":"memcached",
 "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
 "cacheParameterGroup":{
 "cacheParameterGroupName":"default.memcached1.4",
 "cacheNodeIdsToReboot":{
 },
 "parameterApplyStatus":"in-sync"
 },
 "cacheClusterCreateTime":"Dec 1, 2014 10:16:06 PM",
 "preferredAvailabilityZone":"Multiple",
 "numCacheNodes":2,
 "cacheNodeType":"cache.m1.small",
 "cacheClusterStatus":"modifying",
 "autoMinorVersionUpgrade":true,
 "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
 "cacheClusterId":"test-memcached",
 "engineVersion":"1.4.14",
 "cacheSecurityGroups":[
 {
 "status":"active",
 "cacheSecurityGroupName":"default"
 }
],
 "configurationEndpoint":{
 "address":"test-memcached.example.cfg.use1prod.cache.amazonaws.com",
 "port":11211
 },
 "pendingModifiedValues":{
 "numCacheNodes":3
 }
 },
 "requestID":"807f4bc3-354c-11e4-9376-c1d979ba565a",
 "eventID":"e9163565-376f-4223-96e9-9f50528da645"
}
```

```
}
```

## ElastiCache 명령줄 인터페이스 설정

이 섹션은 명령행 도구 실행을 위한 필수 조건, 명령행 도구를 구할 수 있는 위치, 도구 및 환경 설정 방법을 설명하고 도구 사용의 몇몇 일반적인 예를 포함하고 있습니다.

ElastiCache용 AWS CLI로 이동하는 경우에만 이 주제의 지침을 따르십시오.

### Important

Amazon ElastiCache 명령줄 인터페이스(CLI)는 API 버전 2014-09-30 이후의 ElastiCache 개선 사항을 지원하지 않습니다. 명령줄에서 최신 ElastiCache 기능을 사용하려면 [AWS 명령줄 인터페이스](#)를 사용합니다.

### 주제

- [사전 요구사항](#) (p. 230)
- [명령행 도구 얻기](#) (p. 231)
- [도구 설정](#) (p. 231)
- [도구에 대한 자격 증명 제공](#) (p. 232)
- [환경 변수](#) (p. 233)

## 사전 요구사항

이 문서는 Linux/UNIX 또는 Windows 환경에서 작업할 수 있음을 가정합니다. 또한 Amazon ElastiCache 명령줄 도구는 UNIX 기반 환경인 Mac OS X에서도 작동하지만 이 설명서에는 특정 Mac OS X 지침이 포함되어 있지 않습니다.

하나의 규칙으로서 모든 명령줄 텍스트 앞에 일반적인 **PROMPT>** 명령줄 프롬프트가 나옵니다. 머신의 실제 명령행 프롬프트는 다를 수 있습니다. 또한 Linux/UNIX 고유 명령을 표시하기 위해서는 **#** 를, Windows 고유 명령에 대해서는 **c:\>** 을 사용합니다. 명령의 결과인 출력 예는 접두사 없이 그 후에 즉시 표시됩니다.

## Java 런타임 환경

이 설명서에 사용된 명령행 도구를 실행하려면 Java 버전 5 이상이 있어야 합니다. JRE 또는 JDK 설치가 허용됩니다. Linux/UNIX 및 Windows를 포함한 다양한 플랫폼 용도의 JRE를 살펴보고 다운로드하려면 [Java SE Downloads](#)를 참조하십시오.

### Java Home 변수 설정

명령행 도구는 Java 런타임을 찾기 위해 환경 변수(JAVA\_HOME)를 사용합니다. 이 환경 변수는 실행 가능한 bin(Linux 및 UNIX) 또는 java(Windows) 실행 파일을 차례로 포함하고 있는 java.exe이라는 하위 디렉토리가 있는 디렉토리의 전체 경로로 설정되어야 합니다.

Java Home 변수를 설정하려면

1. Java Home 변수를 설정합니다.
  - Linux 및 UNIX에서 다음 명령을 입력합니다.

```
$ export JAVA_HOME=<PATH>
```

- Windows에서 다음 명령을 입력합니다.

```
C:\> set JAVA_HOME=<PATH>
```

2. **\$JAVA\_HOME/bin/java -version**을 실행하고 출력을 확인하여 경로 설정을 확인합니다.

- Linux/UNIX에서 다음과 유사한 출력을 확인할 수 있습니다.

```
$ $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)
```

- Windows에서 다음과 유사한 출력을 확인할 수 있습니다.

```
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)
```

## 명령행 도구 얻기

명령줄 도구는 [ElastiCache 개발자 도구 웹 사이트](#)에서 ZIP 파일로 제공합니다. 이러한 도구는 Java로 작성되었으며 Windows 2000/XP/Vista/Windows 7, Linux/UNIX 및 Mac OSX에 대한 셸 스크립트를 포함하고 있습니다. ZIP 파일은 자체 포함되어 있고 설치가 필요 없으므로 간단히 해당 Zip 파일을 다운로드하여 로컬 머신의 디렉토리에 압축을 풉니다.

## 도구 설정

명령행 도구는 지원 라이브러리를 찾기 위해 환경 변수(AWS\_ELASTICACHE\_HOME)를 사용합니다. 이 환경 변수를 먼저 설정해야 도구를 사용할 수 있습니다. 환경 변수를 명령행 도구의 압축을 푼 디렉토리 경로로 설정합니다. 이 디렉토리 이름은 ElastiCacheCli-A.B.nnnn(A, B 및 n은 버전/릴리스 번호)이며 bin 및 lib라는 하위 디렉토리를 포함하고 있습니다.

AWS\_ELASTICACHE\_HOME 환경 변수를 설정하려면

- 명령행 창을 열고 다음 명령 중 하나를 입력하여 AWS\_ELASTICACHE\_HOME 환경 변수를 설정합니다.
  - Linux 및 UNIX에서 다음 명령을 입력합니다.

```
$ export AWS_ELASTICACHE_HOME=<path-to-tools>
```

- Windows에서 다음 명령을 입력합니다.

```
C:\> set AWS_ELASTICACHE_HOME=<path-to-tools>
```

도구 사용을 좀 더 쉽게 하려면 도구의 BIN 디렉토리를 시스템 경로에 추가하는 것이 좋습니다. 이 설명서의 나머지 부분은 BIN 디렉토리가 시스템 경로에 있음을 가정합니다.

도구의 BIN 디렉토리를 시스템 경로에 추가하려면

- 다음 명령을 입력하여 도구의 BIN 디렉토리를 시스템 경로에 추가합니다.

- Linux 및 UNIX에서 다음 명령을 입력합니다.

```
$ export PATH=$PATH:$AWS_ELASTICACHE_HOME/bin
```

- Windows에서 다음 명령을 입력합니다.

```
C:\> set PATH=%PATH%;%AWS_ELASTICACHE_HOME%\bin
```

#### Note

Windows 환경 변수는 명령 창을 닫으면 재설정됩니다. 이러한 변수를 영구적으로 설정할 수 있습니다. 자세한 내용은 현재 Windows 버전의 설명서를 참조하십시오.

#### Note

경로에 공백이 포함된 경우 다음 예에서처럼 큰따옴표로 묶어야 합니다.  
"C:\Program Files\Java"

## 도구에 대한 자격 증명 제공

명령행 도구는 AWS 계정과 함께 제공되는 AWS 액세스 키 및 보안 액세스 키가 필요합니다. 명령행을 사용하거나 로컬 시스템에 위치하는 자격 증명 파일에서 이러한 키를 얻을 수 있습니다.

배포 파일에는 개별 정보를 사용하여 수정해야 하는 템플릿 파일인 `credential-file-path.template`가 포함되어 있습니다. 다음은 템플릿 파일의 콘텐츠입니다.

```
AWSAccessKeyId=<Write your AWS access ID>
AWSSecretKey=<Write your AWS secret key>
```

#### Important

UNIX에서는 자격 증명 파일의 소유자로 권한을 제한합니다.

```
$ chmod 600 <the file created above>
```

자격 증명 파일 설정을 사용하여 `AWS_CREDENTIAL_FILE` 환경 변수를 설정해야 ElastiCache 도구가 정보를 찾을 수 있습니다.

`AWS_CREDENTIAL_FILE` 환경 변수를 설정하려면

#### 1. 환경 변수 설정:

- Linux 및 UNIX에서는 다음 명령을 사용하여 변수를 업데이트합니다.

```
$ export AWS_CREDENTIAL_FILE=<the file created above>
```

- Windows에서는 다음 명령을 사용하여 변수를 설정합니다.

```
C:\> set AWS_CREDENTIAL_FILE=<the file created above>
```

#### 2. 설정한 것이 제대로 작동하는지 확인하려면 다음 명령을 실행합니다.

```
elasticache --help
```

모든 ElastiCache 명령에 대한 사용법 페이지가 표시됩니다.

## 환경 변수

환경 변수는 스크립팅, 기본값 구성 또는 기본값 임시 재정의에 유용할 수 있습니다.

AWS\_CREDENTIAL\_FILE 환경 변수뿐만 아니라 ElastiCache 명령줄 인터페이스와 함께 포함된 API 도구는 대부분 다음 변수를 지원하지 않습니다.

- EC2\_REGION — 사용할 AWS 리전입니다.
- AWS\_ELASTICACHE\_URL — 서비스 호출에 사용할 URL입니다. EC2\_REGION이 지정되어 있거나 --region 파라미터가 전달되는 경우 다른 리전 엔드포인트를 지정할 필요가 없습니다.

다음 예제에서는 EC2\_REGION이라는 환경 변수를 설정하여 API 도구에서 사용하는 리전을 구성하는 방법을 보여줍니다.

Linux, OS X 또는 Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

## Amazon ElastiCache 오류 메시지

다음 오류 메시지가 Amazon ElastiCache에 의해 반환됩니다. ElastiCache, 기타 AWS 서비스 또는 Memcached에 의해 반환되는 다른 오류 메시지를 받을 수 있습니다. ElastiCache가 아닌 소스의 오류 메시지에 대한 설명은 오류 메시지를 생성하는 소스의 설명서를 참조하십시오.

- 클러스터 노드 할당량 초과 (p. 233)
- 고객의 노드 할당량 초과 (p. 233)
- 부족한 캐시 클러스터 용량 (p. 234)

오류 메시지: 클러스터 노드 할당량이 초과되었습니다. 각 클러스터는 이 리전에서 최대 %n개의 노드를 가질 수 있습니다.

원인: 클러스터를 생성 또는 수정하려고 시도하여 클러스터에 %n개가 넘는 노드가 있습니다.

솔루션: 클러스터에 %n개가 넘는 노드가 있지 않도록 요청을 변경하십시오. 또는 %n 이상의 노드가 필요한 경우 [Amazon ElastiCache 노드 요청 양식](#)을 사용하여 요청하십시오.

자세한 내용은 Amazon Web Services 일반 참조의 [Amazon ElastiCache 한계](#)를 참조하십시오.

오류 메시지: 고객 노드 할당량이 초과되었습니다. 이 리전에서 최대 %n개의 노드가 있을 수 있습니다. 또는 이 리전에서 이미 %s개의 노드 할당량에 도달했습니다.

원인: 클러스터를 생성 또는 수정하려고 시도하여 계정에 이 리전의 모든 클러스터에 대해 %n개가 넘는 노드가 있습니다.

솔루션: 이 계정에 대한 모든 클러스터의 리전에 있는 총 노드가 %n개를 초과하지 않도록 요청을 변경하십시오. 또는 %n 이상의 노드가 필요한 경우 [Amazon ElastiCache 노드 요청 양식](#)을 사용하여 요청하십시오.

자세한 내용은 Amazon Web Services 일반 참조의 [Amazon ElastiCache 한계](#)를 참조하십시오.

오류 메시지: InsufficientCacheClusterCapacity

원인: 현재 AWS에 요청에 대한 서비스를 제공할 수 있을 만큼 온디맨드 용량이 충분하지 않습니다.

솔루션:

- 몇 분 정도 기다린 후 다시 요청을 제출합니다. 용량은 자주 변할 수 있습니다.
- 노드 또는 샤드(노드 그룹) 수가 줄어든 새 요청을 제출하십시오. 예를 들어 단일 요청을 통해 노드 15개를 시작하는 경우 노드 5개의 요청 3개 또는 노드 1개 대신 요청 15개를 시도합니다.
- 클러스터를 시작하고 있는 경우 가용 영역을 지정하지 않고 새 요청을 제출하십시오.
- 클러스터를 시작하고 있는 경우 이후의 단계에서 확장할 수 있는 다른 노드 유형을 사용하여 새 요청을 제출하십시오. 자세한 내용은 [ElastiCache for Memcached 클러스터 조정 \(p. 113\)](#) 단원을 참조하십시오.

## 알림

이 주제에서는 관심을 가질 수 있는 ElastiCache 알림을 다룹니다. 알림은 대부분 일시적이며, 솔루션을 찾아 구현할 때까지만 지속되는 상황이나 이벤트입니다. 알림에는 일반적으로 시작 날짜와 해결 날짜가 있으며, 그 이후에는 알림이 더 이상 관련되지 않습니다. 알림은 사용자와 관련이 있거나 관련이 없을 수 있습니다. 클러스터의 성능을 향상하는 구현 지침을 따르는 것이 좋습니다.

알림은 새로운 또는 향상된 ElastiCache 기능을 소개하지 않습니다.

## 일반 ElastiCache 알림

현재 엔진별로 분류되지 않은 미해결 ElastiCache 알림은 없습니다.

## ElastiCache for Memcached 알림

다음 ElastiCache 알림은 Memcached 엔진에 관한 것입니다.

ElastiCache for Memcached 관련 알림

- [알림: 세분화 결함을 유발하는 Memcached LRU 크롤러 \(p. 234\)](#)

## 알림: 세분화 결함을 유발하는 Memcached LRU 크롤러

알림 날짜: 2017년 2월 28일

일부 환경에서는 클러스터에 Memcached LRU 크롤러의 세분화 결함으로 인한 불안정성이 표시될 수 있습니다. 이는 일정 기간 동안 존재하는 Memcached 엔진 내부의 문제입니다. 이 문제는 LRU 크롤러가 기본적으로 활성화된 Memcached 1.4.33에서 명백하게 드러납니다.

이 문제를 겪고 있는 경우 수정될 때까지 LRU 크롤러를 비활성화하는 것이 좋습니다. 이렇게 하려면 명령줄에서 `lru_crawler disable`을 사용하거나 `lru_crawler` 파라미터 값을 수정(선택됨)하십시오.

해결된 날짜:

해결:

# 설명서 기록

- API 버전: 2015-02-02
- 설명서 최종 업데이트: 2019년 11월 12일

다음 표에서는 2018년 3월 이후 ElastiCache for Memcached 사용 설명서의 각 릴리스에서 변경된 중요 사항에 대해 설명합니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

최신 ElastiCache for Memcached 업데이트

update-history-change	update-history-description	update-history-date
<a href="#">이제 Amazon ElastiCache에서는 T3 표준 캐시 노드가 지원됩니다. (p. 236)</a>	이제 Amazon ElastiCache에서 차세대 범용 순간 확장 가능 T3 표준 캐시 노드를 실행할 수 있습니다. Amazon EC2의 T3 표준 인스턴스는 기존 수준의 CPU 성능과 더불어 누적된 크레딧이 소진될 때까지 언제든지 CPU 사용량을 순간 확장할 수 있는 기능을 제공합니다. 자세한 내용은 <a href="#">지원되는 노드 유형</a> 단원을 참조하십시오.	November 12, 2019
<a href="#">ElastiCache 표준 예약 인스턴스 제품: 부분 선결제, 전체 선결제, 선결제 없음. (p. 236)</a>	예약 인스턴스는 인스턴스 유형 및 AWS 리전에 따라 1년 또는 3년 기간으로 Amazon ElastiCache 인스턴스를 예약할 수 있는 유연성을 제공합니다. 자세한 내용은 <a href="#">예약 노드로 비용 관리</a> 를 참조하십시오.	January 18, 2019
<a href="#">사용 설명서 재구성 (p. 236)</a>	단일 ElastiCache 사용 설명서 가 재구성되어 Redis용 사용 설명서 ( <a href="#">Redis용 ElastiCache 사용 설명서</a> )와 Memcached용 사용 설명서 ( <a href="#">ElastiCache for Memcached 사용 설명서</a> )가 별도로 존재합니다. <a href="#">AWS CLI 명령 참조: elasticache</a> 단원과 <a href="#">Amazon ElastiCache API 참조</a> 의 문서 구조는 그대로 유지됩니다.	April 20, 2018

다음 표에서는 2018년 3월 이전의 ElastiCache for Memcached 사용 설명서에 적용되는 주요 변경 사항에 대해 설명합니다.

변경 사항	설명	변경 날짜
아시아 태평양(오사카-로컬) 지원.	ElastiCache에서 아시아 태평양(오사카-로컬) 리전에 대한 지원이 추가되었습니다. 아시아 태평양(오사카-로컬) 리전은 현재 단일 가용 영역을 지원하고 있으며 초대를 통해서만 이루어집니다. 자세한 내용은 다음 단원을 참조하십시오.	2018년 2월 12일

변경 사항	설명	변경 날짜
	<ul style="list-style-type: none"> <li>지원되는 리전 및 엔드포인트 (p. 31)</li> <li>지원되는 노드 유형 (p. 36)</li> </ul>	
EU(파리) 지원.	<p>ElastiCache에서 EU(파리) 리전에 대한 지원이 추가되었습니다. 자세한 내용은 다음 단원을 참조하십시오.</p> <ul style="list-style-type: none"> <li>지원되는 리전 및 엔드포인트 (p. 31)</li> <li>지원되는 노드 유형 (p. 36)</li> </ul>	2017년 12월 18일
중국(닝샤) 리전 지원	<p>Amazon ElastiCache에서 중국(닝샤) 리전에 대한 지원이 추가되었습니다. 자세한 내용은 다음 자료를 참조하십시오.</p> <ul style="list-style-type: none"> <li>지원되는 리전 및 엔드포인트 (p. 31)</li> <li>지원되는 노드 유형 (p. 36)</li> </ul>	2017년 12월 11일
서비스 연결 역할 지원	<p>ElastiCache의 이번 릴리스에서는 SLR(서비스 연결 역할)에 대한 지원이 추가되었습니다. 자세한 내용은 다음 자료를 참조하십시오.</p> <ul style="list-style-type: none"> <li>Amazon ElastiCache에 서비스 연결 역할 사용 (p. 203)</li> <li>권한 설정(신규 ElastiCache 사용자에게만 해당) (p. 13)</li> </ul>	2017년 12월 7일
R4 노드 유형 지원	<p>이 버전의 ElastiCache에서는 ElastiCache에서 지원하는 모든 AWS 리전에서 R4 노드 유형에 대한 지원이 추가되었습니다. R4 노드 유형을 온디맨드 또는 예약 캐시 노드로 구입할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.</p> <ul style="list-style-type: none"> <li>지원되는 노드 유형 (p. 36)</li> <li>Memcached 노드 유형별 파라미터 (p. 137)</li> </ul>	2017년 11월 20일
연결 패턴 항목	<p>Amazon VPC에서 ElastiCache 클러스터에 액세스할 수 있는 여러 가지 패턴을 설명하는 항목이 ElastiCache 설명서에 추가되었습니다.</p> <p>자세한 내용은 ElastiCache 사용 설명서의 <a href="#">Amazon VPC에 있는 ElastiCache 클러스터에 액세스하기 위한 액세스 패턴 (p. 174)</a> 단원을 참조하십시오.</p>	2017년 4월 24일
Memcached 1.4.34 지원	<p>이전 Memcached 버전의 여러 가지 수정 사항이 통합된 Memcached 버전 1.4.34 지원이 ElastiCache에 추가되었습니다.</p> <p>자세한 내용은 GitHub의 Memcached에서 <a href="#">Memcached 1.4.34 출시 정보</a>를 참조하십시오.</p>	2017년 10월 4일
Memcached 1.4.33 지원	<p>Memcached 버전 1.4.33에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 다음 자료를 참조하십시오.</p> <ul style="list-style-type: none"> <li>Memcached 버전 1.4.33 (p. 28)</li> <li>Memcached 1.4.33 추가 파라미터 (p. 129)</li> </ul>	2016년 20월 12일

변경 사항	설명	변경 날짜
EU 서부(런던) 리전 지원	EU(런던) 리전에 대한 지원이 ElastiCache에 추가되었습니다. 현재는 노드 유형 T2 및 M4만 지원됩니다. 자세한 내용은 다음 자료를 참조하십시오.  <ul style="list-style-type: none"> <li>• <a href="#">지원되는 리전 및 엔드포인트 (p. 31)</a></li> <li>• <a href="#">지원되는 노드 유형 (p. 36)</a></li> </ul>	2016년 13월 12일
캐나다(몬트리올) 리전 지원	캐나다(몬트리올) 리전에 대한 지원이 ElastiCache에 추가되었습니다. 현재는 이 AWS 리전에서 노드 유형 M4 및 T2만 지원됩니다. 자세한 내용은 다음 단원을 참조하십시오.  <ul style="list-style-type: none"> <li>• <a href="#">지원되는 리전 및 엔드포인트 (p. 31)</a></li> <li>• <a href="#">지원되는 노드 유형 (p. 36)</a></li> </ul>	2016년 12월 8일
M4 및 R3 노드 유형 지원	남아메리카(상파울루) 리전의 M4 및 M4 노드 유형과 중국(북경) 지역의 M4 노드 유형에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 다음 자료를 참조하십시오.  <ul style="list-style-type: none"> <li>• <a href="#">지원되는 리전 및 엔드포인트 (p. 31)</a></li> <li>• <a href="#">지원되는 노드 유형 (p. 36)</a></li> </ul>	2016년 11월 1일
미국 동부 2(오하이오) 리전 지원	M4, T2 및 R3 노드 유형을 통한 미국 동부(오하이오) 리전(us-east-2) 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 다음 자료를 참조하십시오.  <ul style="list-style-type: none"> <li>• <a href="#">지원되는 리전 및 엔드포인트 (p. 31)</a></li> <li>• <a href="#">지원되는 노드 유형 (p. 36)</a></li> </ul>	2016년 10월 17일
M4 노드 유형 지원	ElastiCache에서 지원하는 대부분의 리전에서 M4 노드 유형 패밀리에 대한 지원이 ElastiCache에 추가되었습니다. M4 노드 유형을 온디맨드 또는 예약 캐시 노드로 구입할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.  <ul style="list-style-type: none"> <li>• <a href="#">지원되는 노드 유형 (p. 36)</a></li> <li>• <a href="#">Memcached 노드 유형별 파라미터 (p. 137)</a></li> </ul>	2016년 8월 3일
뭄바이 리전 지원	아시아 태평양(뭄바이) 리전에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 다음 자료를 참조하십시오.  <ul style="list-style-type: none"> <li>• <a href="#">지원되는 노드 유형 (p. 36)</a></li> <li>• <a href="#">Memcached 노드 유형별 파라미터 (p. 137)</a></li> </ul>	2016년 6월 27일
R3 노드 유형 지원	중국(베이징) 리전 및 남아메리카(상파울루) 리전의 R3 노드 유형에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 <a href="#">지원되는 노드 유형 (p. 36)</a> 단원을 참조하십시오.	2016년 3월 16일
Lambda 함수를 사용하여 ElastiCache 액세스	Amazon VPC에서 ElastiCache에 액세스하기 위해 Lambda 함수를 구성할 수 있는 자습서가 추가되었습니다. 자세한 내용은 <a href="#">ElastiCache 자습서 및 동영상 (p. 18)</a> 단원을 참조하십시오.	2016년 2월 12일

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서

변경 사항	설명	변경 날짜
아시아 태평양(서울) 리전 지원	t2, m3 및 r3 노드 유형을 통한 아시아 태평양(서울)(ap-northeast-2) 리전 지원이 ElastiCache에 추가되었습니다.	2016년 1월 6일
Memcached 1.4.28 지원	버전 1.4.14 이후의 Memcached 개선 사항을 포함하여 Memcached 버전 1.4.24에 대한 지원이 ElastiCache에 추가되었습니다. 이 릴리스에서는 LRU(가장 오랫동안 사용되지 않음) 캐시 관리에 대한 지원이 백그라운드 작업으로 추가되고 jenkins 또는 murmur3 선택이 해싱 알고리즘, 새로운 명령 및 기타 버그 수정으로 추가됩니다. 자세한 내용은 다음 자료를 참조하십시오. <ul style="list-style-type: none"> <li>• <a href="#">Memcached 출시 정보</a></li> <li>• <a href="#">Memcached 버전 1.4.24 (p. 29)</a>(출처: ElastiCache for Memcached 사용 설명서)</li> </ul>	2015년 8월 27일
PHP 5.6을 사용하는 Memcached 자동 검색 지원	이 Amazon ElastiCache 릴리스에서는 PHP 버전 5.6용 Memcached 자동 검색 클라이언트에 대한 지원이 추가됩니다. 자세한 내용은 <a href="#">PHP용 ElastiCache 클러스터 클라이언트에 대한 소스 코드 컴파일 (p. 64)</a> 단원을 참조하십시오.	2015년 7월 29일
새로운 주제: AWS 외부에서 ElastiCache 액세스	AWS 외부에서 ElastiCache 리소스에 액세스하는 방법에 대한 새로운 항목이 추가되었습니다. 자세한 내용은 <a href="#">AWS 외부에서 ElastiCache 리소스 액세스 (p. 93)</a> 단원을 참조하십시오.	2015년 7월 9일
노드 대체 메시지 추가	ElastiCache에 노드 대체와 관련된 세 가지 메시지(ElastiCache:NodeReplacementScheduled, ElastiCache:NodeReplacementRescheduled 및 ElastiCache:NodeReplacementCanceled)가 추가되었습니다. <p>이에 대한 자세한 내용과 노드 대체를 예약한 경우 취할 수 있는 조치는 ElastiCache의 <a href="#">이벤트 알림 및 Amazon SNS (p. 150)</a> 단원을 참조하십시오.</p>	2015년 6월 11일
비용 할당 태그 지원	비용 할당 태그에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 <a href="#">비용 할당 태그를 사용한 비용 모니터링 (p. 153)</a> 단원을 참조하십시오.	2015년 2월 9일
AWS GovCloud (US-West) 리전 지원	AWS GovCloud (US-West)(us-gov-west-1) 리전에 대한 지원이 ElastiCache에 추가되었습니다.	2015년 1월 29일
유럽(프랑크푸르트) 리전 지원	유럽(프랑크푸르트)(eu-central-1) 리전에 대한 지원이 ElastiCache에 추가되었습니다.	2015년 1월 19일
AWS CloudTrailAPI 호출 로깅 지원	AWS CloudTrail을 사용한 모든 ElastiCache API 호출 로깅에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 <a href="#">AWS CloudTrail을 사용하여 Amazon ElastiCache API 호출 로깅 (p. 226)</a> 단원을 참조하십시오.	2014년 9월 15일
새로운 인스턴스 크기 지원	추가 범용(T2) 인스턴스에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 <a href="#">파라미터 그룹을 사용해 엔진 파라미터 구성 (p. 115)</a> 단원을 참조하십시오.	2014년 9월 11일

Amazon ElastiCache ElastiCache  
for Memcached 사용 설명서

변경 사항	설명	변경 날짜
Memcached에서 유연한 노드 배치 지원	여러 가용 영역에 Memcached 노드를 생성하기 위한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 <a href="#">1 단계: Memcached 클러스터 시작 (p. 14)</a> 단원을 참조하십시오.	2014년 7월 23일
새로운 인스턴스 크기 지원	추가 범용(M3) 인스턴스 및 메모리 최적화(R3) 인스턴스에 대한 지원이 ElastiCache에 추가되었습니다. 자세한 내용은 <a href="#">파라미터 그룹을 사용해 엔진 파라미터 구성 (p. 115)</a> 단원을 참조하십시오.	2014년 7월 1일
PHP 자동 검색	PHP 버전 5.5에 대한 지원이 추가되었습니다. 자세한 내용은 <a href="#">PHP용 ElastiCache 클러스터 클라이언트 설치 (p. 58)</a> 단원을 참조하십시오.	2014년 5월 13일
기본 Amazon Virtual Private Cloud(VPC) 지원	이 릴리스에서 ElastiCache는 Amazon Virtual Private Cloud(VPC)와 완전히 통합되었습니다. 새로운 고객의 경우 캐시 클러스터가 기본적으로 Amazon VPC에 생성됩니다. 자세한 내용은 <a href="#">Amazon VPC 및 ElastiCache 보안 (p. 170)</a> 단원을 참조하십시오.	2013년 1월 8일
캐시 노드 자동 검색을 위한 PHP 지원	캐시 노드 Auto Discovery는 초기 릴리스부터 Java 프로그램에 대한 지원을 제공했습니다. 이 릴리스에서 ElastiCache는 캐시 노드 Auto Discovery를 PHP에 제공합니다.	2013년 1월 2일
Amazon Virtual Private Cloud(VPC) 지원	이 릴리스에서 ElastiCache 클러스터를 Amazon Virtual Private Cloud(VPC)에서 실행할 수 있습니다. 기본적으로 새로운 고객의 캐시 클러스터가 자동으로 Amazon VPC에 생성됩니다. 기존 고객은 자신의 속도로 Amazon VPC로 마이그레이션할 수 있습니다. 자세한 내용은 <a href="#">Amazon VPC 및 ElastiCache 보안 (p. 170)</a> 단원을 참조하십시오.	2012년 20월 12일
캐시 노드 자동 검색 및 새로운 캐시 엔진 버전	ElastiCache는 캐시 노드 Auto Discovery—라는 기능을 제공하는데 이는 클라이언트 프로그램이 클러스터의 모든 캐시 노드를 자동으로 확인하고 이러한 모든 노드에 대한 연결을 시작하고 유지 관리할 수 있도록 합니다.  이 릴리스는 또한 새로운 캐시 엔진 버전인 Memcached 버전 1.4.14를 제공합니다. 이 새 캐시 엔진은 개선된 슬래브 균형 다시 맞추기 기능, 중요한 성능 및 확장성 향상 및 여러 개의 버그 수정을 제공합니다. 여러 가지 새로운 캐시 파라미터를 구성할 수 있습니다. 자세한 내용은 <a href="#">파라미터 그룹을 사용해 엔진 파라미터 구성 (p. 115)</a> 단원을 참조하십시오.	2012년 11월 28일
새로운 캐시 노드 유형	이 릴리스에서는 캐시 노드 유형 4개를 추가로 제공합니다.	2012년 11월 13일
예약 캐시 노드	이 릴리스는 예약된 캐시 노드에 대한 지원을 추가합니다.	2012년 4월 5일
새 가이드	이 설명서는 Amazon ElastiCache 사용 설명서의 첫 번째 배포 버전입니다.	2011년 8월 22일

# AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the AWS General Reference.