



ElastiCache 對於內存緩存用戶指南

# Amazon ElastiCache



API 版本 2015-02-02

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

# Amazon ElastiCache: ElastiCache 對於內存緩存用戶指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

什麼是內 ElastiCache 存緩存？ .....	1
無伺服器快取 .....	1
自行設計的叢集 .....	1
運作方式 .....	2
快取和快取引擎 .....	2
使用案例 .....	7
記憶體內資料存放區 .....	7
ElastiCache 客戶見證 .....	8
ElastiCache for Memcached 資源 .....	8
用於管理實作的工具 .....	10
使用 AWS Management Console .....	10
使用 AWS CLI .....	10
使用 AWS SDK .....	10
使用 ElastiCache API .....	10
另請參閱 .....	10
選擇部署選項 .....	11
比較 Memcached 與 Redis 自行設計快取 .....	12
ElastiCache for Memcached 入門 .....	17
設定 .....	17
註冊一個 AWS 帳戶 .....	17
建立具有管理權限的使用者 .....	18
授與程式設計存取權 .....	19
設定許可 .....	20
設定 EC2 .....	21
授予網路存取權 .....	21
建立快取 .....	22
建立無伺服器快取 .....	22
讀取和寫入資料 .....	23
(選用) 清除 .....	28
後續步驟 .....	29
教學課程：設定 Lambda 函數以在 Amazon VPC 中存取 Amazon ElastiCache .....	30
步驟 1：建立 ElastiCache 快取 .....	30
步驟 2：建立 Lambda 函數 .....	31
步驟 3：測試 Lambda 函數 .....	36

教學和影片 .....	37
影片 .....	38
選擇區域與可用區域 .....	43
可用區域的考量 .....	44
支援的區域與端點 .....	46
安置您的節點 .....	51
使用本地區域 .....	51
啟用本地區域 .....	52
使用 Outposts .....	52
透過 Memcached 主控台使用 Outpost .....	53
使用 Outposts 與 CLI AWS .....	54
自行設計 ElastiCache 叢集 .....	56
ElastiCache 用於內存緩存的組件和功能 .....	57
節點 .....	57
叢集 .....	58
AWS 區域和可用區域 .....	59
端點 .....	60
參數群組 .....	60
安全 .....	60
子網路群組 .....	60
事件 .....	61
管理叢集 .....	61
選擇網路類型 .....	63
資料分層 .....	64
自動探索 .....	65
準備叢集 .....	104
建立叢集 .....	110
檢視叢集的詳細資訊 .....	113
修改叢集 .....	118
重新啟動叢集 .....	121
將節點新增至叢集 .....	122
從叢集中移除節點 .....	128
取消待定的新增或刪除節點作業 .....	133
刪除叢集 .....	134
存取您的叢集 .....	136
尋找連線端點 .....	142

管理節點 .....	148
檢視 ElastiCache 節點狀態 .....	149
連線至節點 .....	154
支援的節點類型 .....	156
替換節點 .....	164
預留節點 .....	166
遷移上一代節點 .....	177
使用 ElastiCache .....	179
快照和還原 .....	179
限制 .....	180
排程自動備份 .....	181
取得手動備份 .....	182
建立最終備份 .....	184
描述備份 .....	186
複製備份 .....	187
從備份進行還原 .....	189
刪除備份 .....	190
標記備份 .....	191
引擎版本和升級 .....	192
支援的 Memcached 版本 .....	193
引擎版本和升級 .....	197
如何升級引擎版本 .....	198
最佳實務和快取策略 .....	199
Memcached 用戶端的最佳實務 .....	199
支援的 Memcached 命令 .....	206
快取策略 .....	207
管理您的自行設計叢集 .....	212
管理維護作業 .....	212
使用參數群組設定引擎參數 .....	214
縮放記憶 ElastiCache 體快取 .....	253
縮放記憶 ElastiCache 體快取 .....	253
設定擴展限制以管理成本 .....	253
使用無伺服器預先調整 ElastiCache .....	253
使用主控台和設定縮放限制 AWS CLI .....	254
ElastiCache 為記憶體快取自行設計的叢集擴充 .....	255
為您的 ElastiCache 資源加上標籤 .....	259

使用標籤監控成本 .....	265
使用 AWS CLI 管理標籤 .....	267
使用 ElastiCache API 管理標籤 .....	270
Amazon ElastiCache Well-Architected Lens .....	272
卓越運作支柱 .....	273
安全支柱 .....	280
可靠性支柱 .....	285
效能效率支柱 .....	289
成本最佳化支柱 .....	297
故障診斷 .....	302
連線問題 .....	302
Redis 的客戶端錯誤 .....	303
疑難排解 ElastiCache 無伺服器中的高延遲 .....	303
疑難排解無伺服器中的節流問題 ElastiCache .....	305
相關主題 .....	306
其他疑難排解步 .....	306
安全群組 .....	306
網路 ACL .....	307
路由表 .....	308
DNS 解析 .....	308
透過伺服器端診斷識別問題 .....	309
網路連線能力驗證 .....	313
網路相關限制 .....	315
CPU 使用率 .....	316
從伺服器端終止連線 .....	319
Amazon EC2 執行個體的用戶端疑難排解 .....	320
解析完成單個請求所花費的時間 .....	321
安全 .....	324
資料保護 .....	324
Amazon ElastiCache 的資料安全性 .....	325
網際網路流量隱私權 .....	333
Amazon VPC 和 ElastiCache 安全性 .....	334
Amazon ElastiCache API 和界面 VPC 端點 (AWS PrivateLink) .....	353
子網路和子網路群組 .....	356
身分和存取權管理 .....	362
物件 .....	363

使用身分驗證 .....	363
使用政策管理存取權 .....	366
Amazon 如何與 IAM 合 ElastiCache 作 .....	368
身分型政策範例 .....	374
故障診斷 .....	376
存取控制 .....	378
管理存取概觀 .....	379
法規遵循驗證 .....	407
其他資訊 .....	408
复原能力 .....	408
減少故障 .....	409
基礎架構安全 .....	411
服務更新 .....	411
管理服務更新 .....	411
日誌記錄和監控 .....	417
無伺服器指標和事件 .....	417
無伺服器指標 .....	417
無伺服器事件 .....	420
自行設計叢集的指標和事件 .....	424
自行設計叢集的指標 .....	424
自行設計的叢集事件 .....	424
監控使用情況 .....	431
Amazon SNS 事件監控 .....	441
使用 AWS CloudTrail 記錄 Amazon ElastiCache API 呼叫 .....	456
CloudTrail 中的 Amazon ElastiCache 資訊 .....	456
了解 Amazon ElastiCache 日誌檔案項目 .....	457
配額 .....	461
參考 .....	462
使用 ElastiCache API .....	462
使用查詢 API .....	462
可用程式庫 .....	465
對應用程式進行疑難排解 .....	466
設定 ElastiCache 的 AWS CLI .....	467
先決條件 .....	467
取得命令列工具 .....	469
設定工具 .....	469

---

為工具提供憑證 .....	470
環境變數 .....	471
錯誤訊息 .....	472
通知 .....	473
一般 ElastiCache 通知 .....	473
ElastiCache for Memcached 通知 .....	473
文檔歷史 .....	475
AWS 詞彙表 .....	487
.....	cdlxxxviii

# 什麼是 Amazon ElastiCache 的內存緩存？

歡迎來到 Amazon ElastiCache 的 Memcached 用戶指南。Amazon ElastiCache 是一種 Web 服務，可讓您輕鬆地在雲端中設定、管理和擴展分散式記憶體內資料存放區或快取環境。它提供高效能、可擴展且符合成本效益的快取解決方案。同時也有助於消除部署和管理分散式快取環境的相關複雜性。

您可以使用兩種 ElastiCache 格式操作 Amazon。您可以開始使用無伺服器快取，或選擇自行設計快取叢集。

## Note

Amazon 的 ElastiCache 工作原理與 Redis 和內存緩存的引擎。使用您有興趣的引擎指南。如果您不確定要使用哪個引擎，請參閱本指南中的[比較 Memcached 與 Redis 自行設計快取](#)。

## 無伺服器快取

ElastiCache 為 Memcached 提供無服務器緩存，從而簡化了為您的應用程式添加和操作基於 Memcached 的緩存。ElastiCache Memcached Serverless 可讓您在一分鐘內建立高可用性快取，並且不需要佈建執行個體或設定節點或叢集。開發人員可以使用 ElastiCache 主控台、SDK 或 CLI 指定快取名稱，以建立無伺服器快取。

ElastiCache 無伺服器也不需要規劃和管理快取容量。ElastiCache 持續監控應用程式所使用的快取記憶體和運算，並自動擴充容量以符合應用程式的需求。ElastiCache 透過抽象化底層快取基礎結構和軟體，為開發人員提供簡單的端點體驗。ElastiCache 以自動且透明的方式管理硬體佈建、監控、節點取代和軟體修補，讓您可以專注於應用程式開發，而不必操作快取。

ElastiCache 對於內存緩存無服務器與 Memcached 1.6 及更高版本兼容。

## ElastiCache 為內存緩存集群設計自己的

如果您需要 ElastiCache 對 Memcached 叢集進行精細控制，您可以選擇使用 ElastiCache。ElastiCache 透過為叢集選擇節點類型、節點數目以及跨 AWS 可用區域的節點放置，可讓您操作節點型叢集。由於 ElastiCache 是全受管服務，因此它會自動管理叢集的硬體佈建、監控、節點取代和軟體修補。

ElastiCache 針對 Memcached 叢集設計您自己的叢集可提供更大的彈性和控制權。例如，您可以根據需要選擇操作具有單一 AZ 可用性或跨 AZ 可用性的叢集。自行設計叢集時，您必須負責正確選擇節點

類型和數量，以確保快取擁有足以因應您應用程式所需的容量。您也可以選擇何時將新軟體修補程式套用於 Memcached 叢集。

在 ElastiCache 為 Memcached 設計自己的設計時，您可以選擇運行 Memcached 1.4 及更高版本。

## 運作方式

在這裡，您可以找到 Memcached 部署的主要組件的概述。ElastiCache

### 快取和快取引擎

快取是可用來儲存快取資料的記憶體內資料存放區。一般而言，您的應用程式會在快取中快取經常存取的資料，以最佳化回應時間。ElastiCache 為 Memcached 提供兩種部署選項：無伺服器 and 自行設計的叢集。請參閱 [選擇部署選項](#)

#### Note

Amazon 的 ElastiCache 工作原理與 Redis 和內存緩存的引擎。使用您有興趣的引擎指南。如果您不確定要使用哪個引擎，請參閱本指南中的 [比較 Memcached 與 Redis 自行設計快取](#)。

#### 主題

- [內存緩存 ElastiCache 的工作原理](#)
- [定價維度](#)

### 內存緩存 ElastiCache 的工作原理

ElastiCache 適用於 Memcached 的無伺服器

ElastiCache Memcached 無伺服器可讓您建立快取，而不必擔心容量規劃、硬體管理或叢集設計。您只需提供快取的名稱，然後就會收到單一端點，您可在 Memcached 用戶端中設定該端點以開始存取快取。

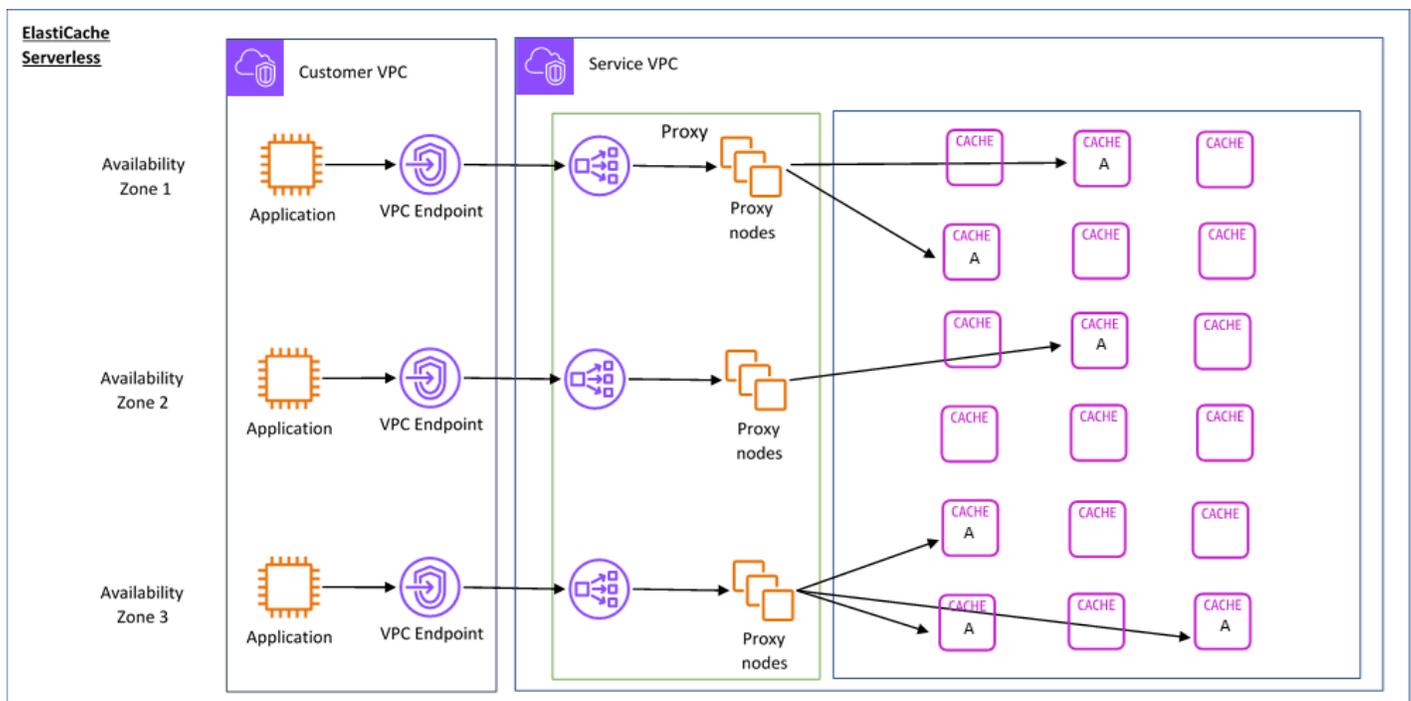
#### Note

ElastiCache 適用於無記憶體快取伺服器僅與支援 TLS 的 Memcached 用戶端相容。

#### 主要優點

- 無需容量規劃：ElastiCache 無伺服器讓您無需規劃容量。ElastiCache 無伺服器會持續監控快取記憶體、運算和網路頻寬使用率，並可垂直和水平調整。它可一邊讓快取節點的大小增加，同時一邊啟動橫向擴展操作，以確保快取隨時可因應您應用程式的需求擴展。
- Pay-per-use：使用 ElastiCache 無伺服器時，您需要支付工作負載在快取中儲存和運算所使用的資料費用。請參閱 [定價維度](#)。
- 高可用性：ElastiCache 無伺服器會自動跨多個可用區域 (AZ) 複製您的資料，以提供高可用性。它會自動監控底層快取節點，並且在發生故障時進行替換。它為每個快取提供 99.99% 可用性 SLA。
- 自動軟體升級：ElastiCache Serverless 會自動將您的快取升級為最新的次要和修補程式軟體版本，而不會影響應用程式的可用性。當有新的 Memcached 主要版本可用時，ElastiCache 會向您發送通知。
- 安全性：Serverless 一律會加密傳輸中的資料和靜態資料。您可以使用服務受管金鑰或使用您自己的客戶受管金鑰來加密靜態資料。

下圖說明 ElastiCache 無伺服器的運作方式。



建立新的無伺服器快取時，請在 VPC 中選擇的子網路中建 ElastiCache 立 Virtual Private Cloud (VPC) (VPC) 端點。您的應用程式可透過這些 VPC 端點連線到快取。

使用 ElastiCache 無伺服器時，您會收到應用程式連線的單一 DNS 端點。當您要求端點的新連線時，ElastiCache 無伺服器會透過 Proxy 層處理所有快取連線。代理層有助於減少複雜的用戶端組態，因為

在基礎叢集發生變更時，用戶端不需要重新探索叢集拓撲。代理層是一組代理節點，可使用 Network Load Balancer 處理連線。當您的應用程式建立新的快取連線時，Network Load Balancer 會將請求傳送至代理節點。當您的應用程式執行快取命令時，連線至應用程式的代理節點會在快取中的快取節點上執行請求。代理層會從用戶端抽象化快取叢集拓撲和節點。這可讓您 ElastiCache 以智慧的方式進行負載平衡、擴充並新增快取節點、在快取節點發生故障時更換快取節點，以及更新快取節點上的軟體，而不會影響應用程式的可用性，也不需要重設連線。

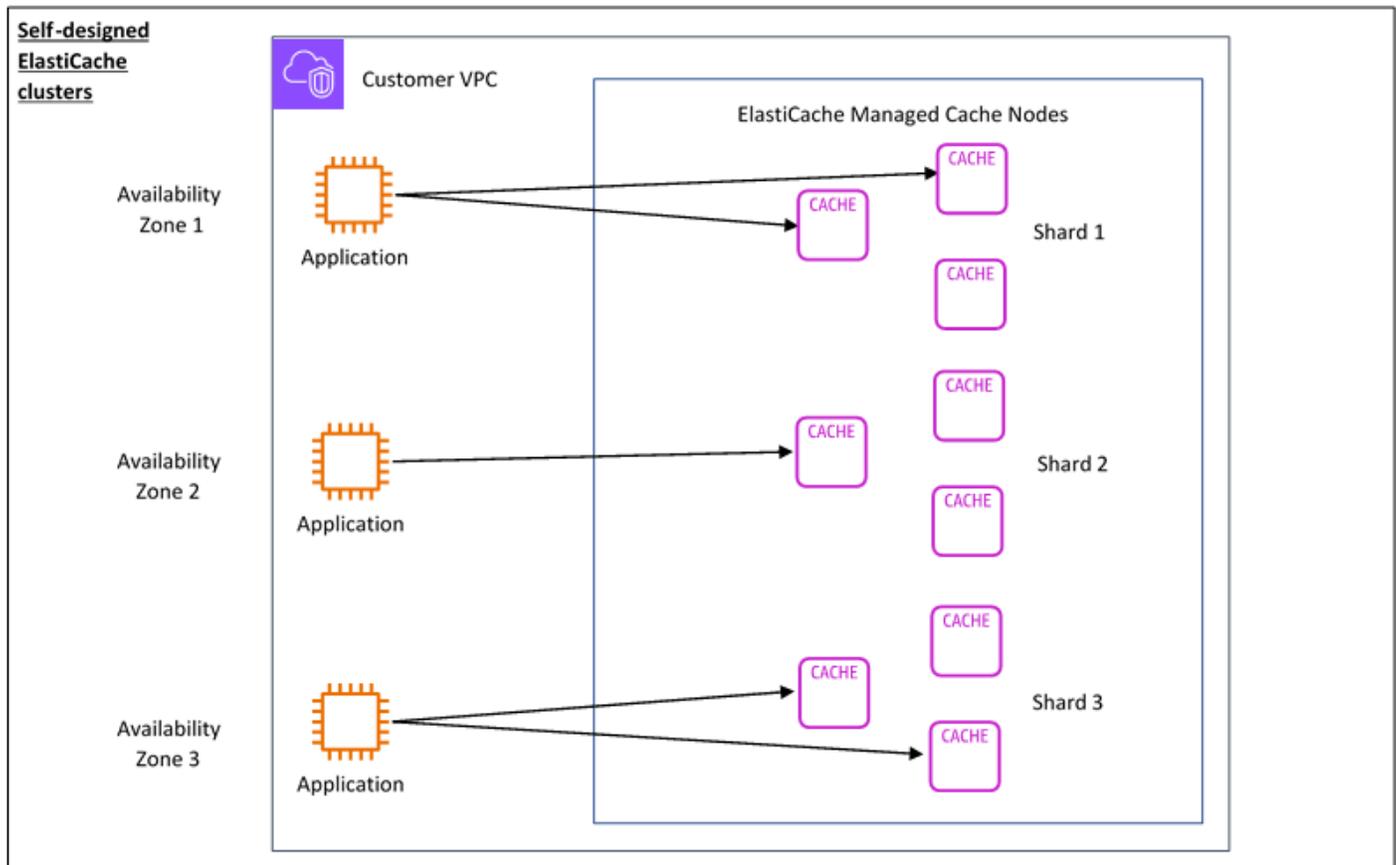
## 自行設計 ElastiCache 的叢集

您可以選擇 ElastiCache 叢集的快取節點系列、大小和節點數目來設計自己的叢集。自行設計叢集能讓您更精準控制叢集的組態和擴展。

### 主要優點

- **設計您自己的叢集：**您可以使用設計自己的叢集 ElastiCache，並選擇要放置快取節點的位置。例如，如果您希望犧牲應用程式的高可用性來換取低延遲，可以選擇在單一 AZ 中部署快取節點。或者，您可以設計讓叢集的節點分布到多個 AZ，以實現高可用性。
- **精細控制：**在自行設計叢集時，您可以透過微調快取的設定來進行更精準的控制。例如，您可以使用 [使用參數群組設定引擎參數](#) 來設定快取引擎。
- **垂直和水平擴展：**您可以選擇在需要時增加或減少快取節點大小來手動擴展叢集。您也可以新增節點來水平擴展。

下圖說明 ElastiCache 自行設計叢集的運作方式。



## 定價維度

您可以 ElastiCache 在兩個部署選項中進行部署。部署 ElastiCache 無伺服器時，您需要支付以 GB 小時為單位儲存的資料使用量，並以 ElastiCache 處理單元 (ECPU) 計算費用。選擇 ElastiCache 針對 Memcached 叢集設計自己的叢集時，您需要支付每小時快取節點使用量的費用。請參閱[此處](#)的定價詳細資訊。

## 資料儲存

您需要支付儲存在 ElastiCache 無伺服器中的資料費用，以 GB-小時 (GB-小時) 計費。ElastiCache 無伺服器會持續監控儲存在快取中的資料、每分鐘多次取樣，並計算每小時平均值，以判斷快取記憶體體的資料儲存使用量 (GB-小時)。每個 ElastiCache 無伺服器快取的計量至少儲存 1 GB 的資料。

## ElastiCache 處理單元

您需要支付應用程式在 ElastiCache 無伺服器上以 ElastiCache 處理單元 (ECPU) 執行的請求付費，該單位包含 vCPU 時間和傳輸的資料。

- 簡單的讀取和寫入傳輸的每 KB 資料需要 1 個 ECPU。例如，最多傳輸 1 KB 資料的 GET 命令會消耗 1 個 ECPU。傳輸 3.2 KB 資料的 SET 請求將會消耗 3.2 個 ECPU。
- 在多個項目上操作的命令將按比例消耗更多 ECPU。例如，如果您的應用程式在 3 個項目上執行 multiget，則會消耗 3 個 ECPU。
- 在更多項目上操作且傳輸更多資料的命令，所消耗的 ECPU 取決於兩個維度中的較高者。例如，如果您的應用程式使用 GET 命令、擷取 3 項目，以及傳輸 3.2 KB 資料，則會消耗 3.2 個 ECPU。或者，如果它只傳輸 2 KB 資料，則會消耗 3 個 ECPU。

ElastiCache 無伺服器會發出名為的新指標，ElastiCacheProcessingUnits 可協助您瞭解工作負載所使用的 ECPU。

### 節點小時數

您可以藉由選擇 EC2 節點系列、大小、節點數，以及在可用區域之間的放置來自行設計快取叢集。自行設計叢集時，您需以每小時為單位支付每個快取節點的費用。

## 常見 ElastiCache 使用案例及 ElastiCache 如何提供協助

不論是提供最新的新聞或產品目錄，或是出售活動門票，速度都是遊戲的重點。傳遞內容的速度，會顯著影響您網站和商業的成功與否。

在「[對於耐心不足的 Web 使用者來說，一眨眼的等待時間也嫌太長](#)」報導中，New York Times 指出，使用者可以注意到競爭網站之間 250 毫秒 (1/4 秒) 的差異。使用者傾向於放棄速度較慢的網站，而選擇速度較快的網站。Amazon 完成的測試，引自[網頁載入時間與訪客之間的關聯](#)，指出每載入 100 毫秒 (1/10 秒) 的負載時間，銷售會下降 1%。

當某人想要資料，若這些資料已經過快取，您就可以更迅速提供資料。無論是網頁還是影響商業決策的報告，都適用同樣的道理。您的企業是否能快取您的網頁、以最短的延遲將其傳遞？

很顯然，需要最大的項目，就是您最想要快取的。但為何不快取較不常用的項目？即使是經過最佳化的資料庫查詢或遠端 API 呼叫，也會比從記憶體內快取擷取一般金鑰慢很多。「明顯較慢」這件事，就會造成客戶流失。

下列範例說明 ElastiCache 能提升您應用程式整體效能的一些方式。

### 記憶體內資料存放區

記憶體內鍵/值存放區的主要目的，是提供超快速 (亞毫秒級延遲) 和經濟實惠的資料複本存取。大多數資料存放區，都具有經常存取但不常更新的資料區域。此外，對資料庫進行查詢，總是比在鍵/值對快取中找出鍵要更慢，且費用更高。執行某些資料庫查詢的費用特別昂貴。其中一個例子是涉及跨多個資料表之聯結的查詢，或具有密集型計算的查詢。透過快取這類查詢結果，您只需支付查詢的一次性費用。然後就可以快速擷取資料多次，而無需重新執行查詢。

### 我應該快取什麼？

在決定要快取哪些資料時，請考慮下列因素：

**速度和費用** - 從資料庫取得資料，總是比從快取要來得慢，且費用更高。某些資料庫查詢在本質上比其他查詢更慢且更昂貴。例如，在多個資料表上執行聯結查詢，與簡單的單一資料表查詢相比，前者的速度明顯較慢且費用更高。如果需要以較慢且費用較高的查詢方式來取得所需資料，則適合改為快取。如果需要以相對快速和簡單的查詢來取得資料，仍可能適合使用快取，具體取決於其他因素。

**資料和存取模式** - 判斷要快取的內容也涉及了解資料本身及其存取模式。例如，快取快速變動或很少存取的資料並沒有意義。若要讓快取提供真正的益處，資料應為相對靜態且經常存取。例如社群媒體網站上的個人資料。相反地，如果快取資料不能提供速度或成本優勢，則不建議快取資料。例如，快取會傳回搜尋結果的網頁並沒有意義，因為查詢和結果通常都是獨一無二的。

過時 - 根據定義，快取的資料是過時的資料。即使某些情況下並非過時，仍應該一律視為過時。若要判斷您的資料是否適合快取，您需要判斷應用程式對過時資料的容錯能力。

您的應用程式或許能承受某個內容中的過時資料，但不能承受另一個內容中的過時資料。例如，假設您的網站提供公開交易股票價格。在附有免責聲明，表示可能有 n 分鐘延遲的情況下，您的客戶可能會接受一定程度的過時性。但是，如果是向銷售或購買的經紀人提供股票價格，您需要即時的資料。

如果下列描述成立，便可考慮快取您的資料：

- 與快取擷取相比，取得您資料的速度緩慢或費用高昂。
- 使用者經常存取您的資料。
- 您的資料相對保持無變動，或者資料會快速變動但過時性不夠成大問題。

如需詳細資訊，請參閱下列內容：

- ElastiCache for Memcached 使用者指南中的[快取策略](#)

## ElastiCache 客戶見證

若要了解 Airbnb、PBS、Esri 和其他企業如何使用 Amazon ElastiCache，透過改善客戶體驗來促進業務成長，請參閱[其他人如何使用 Amazon ElastiCache](#)。

您也可以觀看其他 ElastiCache 客戶使用案例的[教學課程影片](#)。

## ElastiCache for Memcached 資源

建議您在開始前先閱讀以下章節，並在需要時用做參考：

- 服務特色和定價 - [產品詳細資訊頁面](#)提供 ElastiCache 的一般產品概觀、服務特色和定價資訊。
- ElastiCache 影片 - 「[ElastiCache 影片](#)」一節中的影片為您介紹 Amazon ElastiCache for Memcached、說明常用案例，並示範如何使用 ElastiCache for Memcached 降低延遲並改善應用程式輸送量。
- 入門 - 「[Amazon ElastiCache for Memcached 入門](#)」一節包含將引導您完成建立快取叢集、授權存取快取叢集、連線至快取節點以及刪除快取叢集的程序範例。
- 大幅提高效能 - [使用 Amazon ElastiCache 大幅提升效能](#)白皮書介紹可讓應用程式大幅提升效能的快取策略。

如果您希望使用 AWS Command Line Interface (AWS CLI)，這些文件可協助您開始使用：

- [AWS Command Line Interface 文件](#)

本節提供的資訊包含下載 AWS CLI、讓 AWS CLI 在您的系統上運作以及提供您的 AWS 憑證。

- [ElastiCache 的 AWS CLI 說明文件](#)

這份獨立文件說明所有 ElastiCache AWS CLI 的命令，包括語法和範例。

您可以將應用程式編寫為可使用適用各種常用程式設計語言的 ElastiCache API。以下是一些可用資源：

- [適用於 Amazon Web Services 的工具](#)

Amazon Web Services 提供多種支援 ElastiCache for Memcached 的軟體開發套件 (SDK)。您可以使用 Java、.NET、PHP、Ruby 和其他語言來編寫 ElastiCache 程式碼。透過將傳給 ElastiCache 的請求格式化、剖析回應並提供重試邏輯和錯誤處理，這些軟體開發套件可以大幅簡化您的應用程式開發程序。

- [使用 ElastiCache API](#)

如果您不希望使用 AWS 軟體開發套件，可以使用查詢 API 直接和 ElastiCache 互動。您可以在本節中找到建立與驗證請求與處理回應的故障診斷秘訣和資訊。

- [Amazon ElastiCache API 參考](#)

這份獨立文件說明所有 ElastiCache API 作業，包括語法和範例。

## 用於管理實作的工具

為 ElastiCache 叢集授予 Amazon EC2 執行個體存取權後，您就能透過四種方法管理 ElastiCache 叢集：AWS Management Console、ElastiCache 的 AWS CLI、適用於 ElastiCache 的 AWS 軟體開發套件及 ElastiCache API。

### 使用 AWS Management Console

AWS Management Console 是管理 Amazon ElastiCache for Memcached 最簡單的方式。主控台可讓您建立快取叢集、新增和移除快取節點與執行其他管理任務，無需編寫任何程式碼。主控台也提供來自 CloudWatch 的快取節點效能圖表，可顯示快取引擎活動、記憶體和 CPU 用量以及其他指標。如需詳細資訊，請參閱此《使用者指南》中的特定主題。

### 使用 AWS CLI

您也可以使用 ElastiCache 的 AWS Command Line Interface (AWS CLI)。AWS CLI 可讓您輕鬆一次執行一個操作，例如啟動或停止您的快取叢集。您也可以從您選擇的指令碼語言叫用 ElastiCache 的 AWS CLI 命令，以便自動執行重複任務。如需 AWS CLI 的詳細資訊，請參閱使用者指南和 [AWS CLI 命令參考](#)。

### 使用 AWS SDK

如果您希望從應用程式存取 ElastiCache，可以使用其中一個 AWS 軟體開發套件 (SDK)。軟體開發套件包裝 ElastiCache API 呼叫，並讓應用程式與 ElastiCache API 的低階詳細資訊隔離。您將提供您的登入資料，而軟體開發套件程式庫負責身分驗證和請求簽署。如需使用 AWS 軟體開發套件的詳細資訊，請參閱[適用於 Amazon Web Services 的工具](#)。

### 使用 ElastiCache API

您也可以直接針對 ElastiCache Web 服務 API 編寫應用程式程式碼。使用 API 時，您必須編寫必要的程式碼以建構和驗證您的 HTTP 請求、剖析來自 ElastiCache 的結果並處理任何錯誤。如需 API (匯入 API) 的詳細資訊，請參閱「[使用 ElastiCache API](#)」。

### 另請參閱

如需更多管理 Amazon ElastiCache for Memcached 部署的詳細資訊，請參閱下列內容：

- [使用 ElastiCache](#)
- [網際網路流量隱私權](#)

- [在 Amazon ElastiCache 中記錄和監控](#)

## 選擇部署選項

Amazon ElastiCache 提供 2 個部署選項：

- 無伺服器快取
- 自行設計叢集

### 無伺服器快取

Amazon ElastiCache Serverless 可簡化快取建立，並立即擴展以支援客戶要求最嚴苛的應用程式。使用 ElastiCache Serverless 可讓您在一分鐘內建立高度可用且可擴充的快取，而不需要佈建、規劃及管理快取叢集容量。ElastiCache Serverless 會自動以備援方式將資料儲存在三個可用區域之間，並提供 99.99% 可用性 [服務水準協議 \(SLA\)](#)。ElastiCache 提供跨 AZ 的自動資料複寫功能，您不需要手動管理複本和自訂軟體也能讓它們保持同步。

### 自行設計的叢集

如果您需要微調對 ElastiCache for Memcached 叢集的控制，可以選擇透過 ElastiCache 設計自己的 Memcached 叢集。ElastiCache 可讓您藉由為叢集選擇節點類型、節點數和 AWS 可用區域內的節點放置，來操作節點型叢集。由於 ElastiCache 是一項全受管服務，因此可自動管理叢集的硬體佈建、監控、節點替換和軟體修補。

### 選擇部署選項

下列情況請選擇 Serverless 快取：

- 您要為新的或未知的工作負載建立新快取
- 您有無法預期的應用程式流量
- 您希望以最簡單的方式開始使用快取

下列情況請選擇自行設計 ElastiCache 叢集：

- 您已在執行 ElastiCache Serverless，並且希望更精準控制執行 Memcached 的節點類型、節點數，以及節點的放置。
- 您不希望應用程式流量波動太大，或者您可以準確預測應用程式流量的峰值和低谷。
- 您可以預測容量需求以控制成本。

相關主題:

- [自行設計和管理 Memcached 實作的 ElastiCache 叢集](#)

## 比較 Memcached 與 Redis 自行設計快取

Amazon ElastiCache 支持內存緩存和 Redis 的緩存引擎。每個引擎都提供一些優點。使用本主題中的資訊來協助您選擇最符合您需求的引擎和版本。

### Important

建立快取、自行設計的叢集或複寫群組之後，您可以升級至較新的引擎版本，但無法降級至較舊的引擎版本。如果您想要使用較舊的引擎版本，則必須刪除現有的快取、自行設計的叢集或複寫群組，然後使用舊版引擎重新建立。

這兩個引擎表面上看起來很類似。每個引擎都是記憶體內金鑰/值存放區。不過，實際上有顯著的差異。

如果適用下列情況，請選擇 Memcached：

- 您需要盡可能最簡單的模型。
- 您需要執行具有多個核心或執行緒的大型節點。
- 您需要能夠向外和向內擴展，並視您系統需求增加和減少來新增和移除節點。
- 您需要快取物件。

如果您適用 ElastiCache 於以下條件，請選擇具有 Redis 版本的 Redis：

- ElastiCache 適用於版本 7.0 (增強版)

您想要使用 [Redis 函數](#)、[碎片發佈/訂閱](#)或 [Redis ACL 改善](#)功能。如需詳細資訊，請參閱 [Redis 7.0 版 \(增強版\)](#)。

- ElastiCache 對於 Redis 版本 6.2 (增強)

您希望能夠使用 r6gd 節點類型在記憶體和 SSD 之間將資料分層。如需詳細資訊，請參閱[資料分層](#)。

- ElastiCache 對於雷迪斯版本 6.0 (增強)

您想要使用角色型存取控制驗證使用者。

如需詳細資訊，請參閱 [Redis 6.0 版 \(增強版\)](#)。

- ElastiCache 對於雷迪斯版本 5.0.0 ( 增強 )

您想要使用 [Redis 串流](#)，這是一種日誌資料結構，允許製作者即時附加新項目，同時允許消費者以封鎖或非封鎖方式取用訊息。

如需詳細資訊，請參閱 [Redis 5.0.0 版 \(增強版\)](#)。

- ElastiCache 對於雷迪斯版本 4.0.10 ( 增強 )

支援加密，以及從您的 Redis (啟用叢集模式) 叢集動態新增或移除碎片。

如需詳細資訊，請參閱 [Redis 4.0.10 版 \(增強版\)](#)。

以下版本已棄用、已達到或即將達到生命週期結束。

- ElastiCache 對於雷迪斯版本 3.2.10 ( 增強 )

支援從您的 Redis (啟用叢集模式) 叢集動態新增或移除碎片的功能。

 Important

目前 ElastiCache 對於 Redis 的 3.2.10 不支持加密。

如需詳細資訊，請參閱下列內容：

- [Redis 3.2.10 版 \(增強版\)](#)
- Redis 的線上重新分片最佳實務；如需詳細資訊，請參閱下列主題：
  - [最佳實務：線上重新分片](#)
  - [Redis \(啟用叢集模式\) 的線上重新分片和碎片重新平衡功能](#)
- 如需擴展 Redis 叢集的詳細資訊，請參閱[擴展](#)。
- ElastiCache 對於雷迪斯版本 3.2.6 ( 增強 )

如果您需要舊版 Redis 的功能以及下列功能，請 ElastiCache 針對 Redis 3.2.6 選擇：

- ~~傳輸中加密。如需詳細資訊，請參閱 [ElastiCache 適用於 Redis 傳輸中加密的 Amazon](#)。~~

- 靜態加密。如需詳細資訊，請參閱 [ElastiCache 適用於 Redis 閒置加密的 Amazon](#)。
- ElastiCache 對於 Redis ( 群集模式已啟用 ) 3.2.4 版

如果您需要 Redis 2.8.x 的功能及下列功能，請選擇 Redis 3.2.4 (叢集模式)：

- 您需要將資料分割到 2 至 500 個節點群組中 (僅限叢集模式)。
  - 您需要地理空間索引 (叢集模式或非叢集模式)。
  - 您不需要支援多個資料庫。
- ElastiCache 適用於 Redis (非叢集模式) 2.8.x 和 3.2.4 (增強型)

如果適用下列情況，請選擇 Redis 2.8.x 或 Redis 3.2.4 (非叢集模式)：

- 您需要複雜資料類型，例如字串、雜湊、清單、集合、有序集合及點陣圖。
- 您需要排序或排名記憶體內資料集。
- 您的金鑰存放區必須具備持久性。
- 您需要針對讀取密集型應用程式，將資料從主要複本複寫到一或多個僅供讀取複本。
- 您需要在主要節點失敗時自動容錯移轉。
- 您需要發佈和訂閱 (pub/sub) 功能，用於通知用戶端伺服器上的事件。
- 您需要自行設計的叢集以及無伺服器快取的備份和還原功能。
- 您需要支援多個資料庫。

## Memcached、Redis (停用叢集模式) 和 Redis (啟用叢集模式) 的比較摘要

	Memcached	Redis (停用叢集模式)	Redis (啟用叢集模式)
引擎版本 +	1.4.5 及更新版本	4.0.10 和更新版本	4.0.10 和更新版本
資料類型	簡單	2.8.x - 複雜 * 複雜	3.2.x 和更新版本 - 複雜
資料分割	是	否	是
叢集可修改	是	是	3.2.10 和更新版本 - 有限
線上重新分片	否	否	3.2.10 版和更新版本
加密	運輸途中 1.6.12 及更新版本	4.0.10 和更新版本	4.0.10 和更新版本
資料分層	否	6.2 及更高版本	6.2 及更高版本
合規認證			
合規認證			
FedRAMP	是 - 1.6.12 和更新版本	4.0.10 和更新版本	4.0.10 和更新版本
HIPAA	是 - 1.6.12 和更新版本	4.0.10 和更新版本	4.0.10 和更新版本
PCI DSS	是	4.0.10 和更新版本	4.0.10 和更新版本
多執行緒	是	否	否
節點類型升級	否	是	是
引擎升級	是	是	是
高可用性 (複寫)	否	是	是
自動容錯移轉	否	選用	必要
發佈/訂閱功能	否	是	是

	Memcached	Redis (停用叢集模式)	Redis (啟用叢集模式)
有序集合	否	是	是
備份和還原	僅適用於無伺服器 Memcached，不適用於自行設計的 Memcached 叢集	是	是
地理空間索引	否	4.0.10 和更新版本	是

備註：

字串、物件 (例如資料庫)

\* 字串、集合、有序集合、清單、雜湊、點陣圖、hyperloglog

字串、集合、有序集合、清單、雜湊、點陣圖、hyperloglog、地理空間索引

+ 排除已棄用、已到達或即將結束的版本。

選擇適用於您叢集的引擎之後，我們建議您使用該引擎的最新版本。如需詳細資訊，請參閱 [ElastiCache 支援 Memcached 版本](#) 或 [Redis 版本 ElastiCache 支援](#)。

# Amazon ElastiCache for Memcached 入門

本節中的主題會引導您完成使用 ElastiCache 主控台來建立、授權存取、連線至 Memcached 無伺服器快取，以及最後刪除此快取的程序。

## 主題

- [設定](#)
- [步驟 1：建立快取](#)
- [步驟 2：對快取讀取和寫入資料](#)
- [步驟 3：\(選用\) 清除](#)
- [後續步驟](#)
- [教學課程：設定 Lambda 函數以在 Amazon VPC 中存取 Amazon ElastiCache](#)
- [ElastiCache 教學和影片](#)

## 設定

若要設定 ElastiCache：

### 主題

- [註冊一個 AWS 帳戶](#)
- [建立具有管理權限的使用者](#)
- [授與程式設計存取權](#)
- [設定您的權限 \(僅限新 ElastiCache 使用者\)](#)
- [設定 EC2](#)
- [授予從 Amazon VPC 安全群組透過網路存取快取的權限](#)

## 註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 [root 使用者來執行需要 root 使用者存取權](#)的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 <https://aws.amazon.com/> 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

## 建立具有管理權限的使用者

註冊後，請保護您的 AWS 帳戶 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立系統管理使用者，這樣您就不會將 root 使用者用於日常工作。

保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。[AWS Management Console](#)在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的[以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的[為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM 身分中心中，將管理存取權授予使用者。

[若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。](#)

以具有管理權限的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

如需使用 IAM 身分中心使用者 [登入的說明](#)，請參閱 [使用AWS 登入者指南中的登入 AWS 存取入口網站](#)。

## 指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[建立權限集](#)」。

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[新增群組](#)」。

## 授與程式設計存取權

如果使用者想要與 AWS 之外的 AWS Management Console 授與程式設計存取權的方式取決於正在存取的使用者類型。

若要授與使用者程式設計存取權，請選擇下列其中一個選項。

哪個使用者需要程式設計存取權？	到	By
人力身分 (IAM Identity Center 中管理的使用者)	使用臨時登入資料來簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	<p>請依照您要使用的介面所提供的指示操作。</p> <ul style="list-style-type: none"> <li>• 如需詳細資訊 AWS CLI，請參閱 <a href="#">《使 AWS CLI 用AWS Command Line Interface者指南》</a> AWS IAM Identity Center 中的〈配置使用〉。</li> <li>• 如需 AWS SDK、工具和 AWS API，請參閱 AWS SDK 和工具參考指南中的 <a href="#">IAM 身分中心身分驗證</a>。</li> </ul>

哪個使用者需要程式設計存取權？	到	By
IAM	使用臨時登入資料來簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	遵循《IAM <a href="#">使用者指南</a> 》中的 <a href="#">〈將臨時登入資料搭配 AWS 資源使用〉</a> 中的指示
IAM	(不建議使用) 使用長期認證簽署對 AWS CLI、AWS SDK 或 AWS API 的程式設計要求。	請依照您要使用的介面所提供的指示操作。 <ul style="list-style-type: none"> <li>• 如需相關資訊 AWS CLI，請參閱使用指南中的<a href="#">使用 IAM 使用者登入資料進行驗證</a>。AWS Command Line Interface</li> <li>• 對於 AWS SDK 和工具，請參閱 AWS SDK 和工具參考指南中的<a href="#">使用長期憑據進行身份驗證</a>。</li> <li>• 如需 AWS API，請參閱 IAM 使用者指南中的<a href="#">管理 IAM 使用者的存取金鑰</a>。</li> </ul>

#### 相關主題:

- IAM 使用者指南中的[什麼是 IAM？](#)。
- AWS AWS 一般參考中的[安全登入資料](#)。

## 設定您的權限 (僅限新 ElastiCache 使用者)

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 使用者和群組位於 AWS IAM Identity Center：

建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 [建立權限合集](#) 說明進行操作。

- 透過身分提供者在 IAM 中管理的使用者：

建立聯合身分的角色。請按照 IAM 使用者指南的 [為第三方身分提供者 \(聯合\) 建立角色](#) 中的指示進行操作。

- IAM 使用者：
  - 建立您的使用者可擔任的角色。請按照 IAM 使用者指南的 [為 IAM 使用者建立角色](#) 中的指示進行操作。
  - (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的 [新增許可到使用者 \(主控台\)](#) 中的指示。

Amazon ElastiCache 建立並使用服務連結角色來佈建資源，並代表您存取其他 AWS 資源和服務。ElastiCache 若要為您建立服務連結角色，請使用名為的 AWS-managed 策略。AmazonElastiCacheFullAccess 此角色隨附了預先佈建、服務代表您建立服務連結角色所需的許可。

您可能決定不使用預設的政策，而是改為使用自訂的受管政策。在此情況下，請確定您具有呼叫權限，iam:createServiceLinkedRole 或已建立 ElastiCache 服務連結角色。

如需詳細資訊，請參閱下列內容：

- [建立新政策 \(IAM\)](#)
- [Amazon ElastiCache 的 AWS 受管政策](#)
- [為 Amazon ElastiCache 使用服務連結角色](#)

## 設定 EC2

您需要設定 EC2 執行個體，以便從中連線至您的快取。

- 如果您還沒有 EC2 執行個體，請在此處了解如何設定 EC2 執行個體：[開始使用 EC2](#)。
- EC2 執行個體必須與快取位於相同的 VPC 中，並且具有相同的安全群組設定。依預設，Amazon ElastiCache 會在您的預設 VPC 中建立快取，並使用預設的安全群組。若要依照本教學課程進行，請確認您的 EC2 執行個體位於預設 VPC 中，並且具有預設安全群組。

## 授予從 Amazon VPC 安全群組透過網路存取快取的權限

ElastiCache 對於內存緩存使用 11211 和 11212 端口來接受內存緩存命令。為了從 EC2 執行個體成功連線和執行 Memcached 命令，您的安全群組必須允許存取這些連接埠。

1. 登入 AWS Command Line Interface 並開啟 [Amazon EC2 主控台](#)。
2. 在導覽窗格中，在 Network & Security (網路與安全) 下，選擇 Security Groups (安全群組)。
3. 從安全群組的清單中，選擇要用於 Amazon VPC 的安全群組。除非您建立了要 ElastiCache 使用的安全性群組，否則此安全性群組會命名為 default。
4. 選擇「傳入」索引標籤，然後：
  - a. 選擇編輯。
  - b. 選擇新增規則。
  - c. 在「類型」欄中，選擇自訂 TCP 規則。
  - d. 在連接埠範圍方塊中，輸入 11211。
  - e. 在來源方塊中，選擇具有連接埠範圍 (0.0.0.0/0) 的隨處，讓任何您在 Amazon VPC 內啟動的 Amazon EC2 執行個體都能連線至您的快取。
  - f. 如果您使用的是 ElastiCache 無伺服器，請選擇「新增規則」來新增其他規則。
  - g. 在 Type (類型) 欄中，選擇 Custom TCP rule (自訂 TCP 規則)。
  - h. 在連接埠範圍方塊中，輸入 11212。
  - i. 在來源方塊中，選擇具有連接埠範圍 (0.0.0.0/0) 的隨處，讓任何您在 Amazon VPC 內啟動的 Amazon EC2 執行個體都能連線至您的快取。
  - j. 選擇儲存

## 步驟 1：建立快取

您即將啟動的快取將會即時執行，而非在沙盒中執行。在刪除快取之前，您都需針對快取支付標準 ElastiCache 使用費。如果您一鼓作氣完成這裡所述的練習，並在完成時刪除快取，則總計費用會很少 (通常不到 1 美元)。如需 ElastiCache 使用費率的詳細資訊，請參閱 [Amazon ElastiCache](#)。

## 建立無伺服器快取

### AWS Management Console

使用 ElastiCache 主控台建立新快取：

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在主控台左側的導覽窗格中，選擇 Memcached 快取。
3. 在主控台的右側，選擇建立 Memcached 快取。

4. 在快取設定中，輸入名稱。您可以選擇性地輸入快取的說明。
5. 讓預設設定保持選取狀態。
6. 按一下建立以建立快取。
7. 快取處於「作用中」狀態後，您就可以開始對快取寫入和讀取資料。

### 使用 AWS CLI 建立新快取

下列 AWS CLI 範例會使用 `create-serverless-cache` 建立新快取。

#### Linux

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

#### Windows

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

請注意，「狀態」欄位的值會設定為 `CREATING`。

若要確認 ElastiCache 已完成建立快取，請使用 `describe-serverless-caches` 命令。

#### Linux

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

#### Windows

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

建立新快取後，繼續進行 [步驟 2：對快取讀取和寫入資料](#)。

## 步驟 2：對快取讀取和寫入資料

本節假設您已建立 Amazon EC2 執行個體且可連線至該執行個體。如需操作方式說明，請參閱 [Amazon EC2 入門指南](#)。

依預設，ElastiCache 會在預設 VPC 中建立快取。請確認您的 EC2 執行個體同樣是在預設 VPC 中建立，如此它才能連線至快取。

## 組態

開始之前，請確定您有正確的連接埠可供存取。

主要連接埠：

讀取最佳化連接埠：11212

無伺服器 Memcached 快取會通告具有相同主機名稱的兩個連接埠。主要連接埠允許以與 OSS Memcached 相同的一致性保證進行寫入和讀取。讀取最佳化的連接埠允許寫入，並且還可以降低延遲的偶發一致讀取。

尋找您的快取端點

## AWS Management Console

若要使用 ElastiCache 主控台尋找快取的端點：

1. 登錄到 AWS Management Console 並打開 Amazon ElastiCache 控制台 <https://console.aws.amazon.com/elasticache/>。
2. 在主控台左側的導覽窗格中，選擇 Memcached 快取。
3. 在控制台的右側，按一下您剛建立的快取名稱。
4. 在快取詳細資訊中，尋找並複製快取端點。

## AWS CLI

下列 AWS CLI 範例示範如何使用 `describe-serverless-caches` 命令尋找新快取的端點。執行命令後，查詢「端點」欄位。

## Linux

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

## Windows

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

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

## 使用 OpenSSL 進行連線

如需如何使用 OpenSSL 進行連線的詳細資訊，請參閱 [ElastiCache 傳輸中加密 \(TLS\)](#)。

## 使用 Memcached Java 用戶端進行連線

如需如何使用 Memcached Java 用戶端進行連線的詳細資訊，請參閱 [ElastiCache 傳輸中加密 \(TLS\)](#)

## 使用 Memcached PHP 用戶端進行連線

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

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

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

## 使用 Memcached Python 用戶端 (Pymemcache) 進行連線

請參閱 [https://pymemcache.readthedocs.io/en/latest/getting\\_started.html](https://pymemcache.readthedocs.io/en/latest/getting_started.html)

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

context = ssl.create_default_context()
cluster_endpoint = <To be taken from the AWS CLI / console>
```

```
target_port = 11211
memcached_client = Client("{cluster_endpoint}", target_port, tls_context=context)
memcached_client.set("key", "value", expire=500, noreply=False)
assert self.memcached_client.get("key").decode() == "value"
```

## 使用 Memcached NodeJS/TS 用戶端 (Electrode-IO memcache) 進行連線

請參閱 <https://github.com/electrode-io/memcache> 和 <https://www.npmjs.com/package/memcache-client>

透過 `npm i memcache-client` 進行安裝

在應用程式中，依照下述建立 Memcached TLS 用戶端：

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

## 使用 Memcached Rust 用戶端 (rust-memcache) 進行連線

請參閱 <https://crates.io/crates/memcache> 和 <https://github.com/aisk/rust-memcache>。

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

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

## 使用 Memcached Go 用戶端 (Gomemcache) 進行連線

請參閱 <https://github.com/bradfitz/gomemcache>

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

```
foo := &Item{Key: "foo", Value: []byte("fooval"), Flags: 123}
err := c.Set(foo)
```

## 使用 Memcached Ruby 用戶端 (Dalli) 進行連線

請參閱 <https://github.com/petergoldstein/dalli>

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

## 使 Connect 記憶體快取的 .NET 用戶端連線 () EnyimMemcachedCore

請參閱 <https://github.com/cnblogs/EnyimMemcachedCore>

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

現在您可以繼續進行 [步驟 3 : \(選用\) 清除](#)。

## 步驟 3 : (選用) 清除

### 使用 AWS Management Console

以下程序會從您的部署中刪除單一快取。若要刪除多個快取，請針對每個要刪除的快取重複此程序。您不需要等待某個快取完成刪除，即可開始刪除其他快取。

### 刪除快取

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 Amazon ElastiCache 主控台。
2. 在 ElastiCache 主控台儀表板中，選擇您要刪除之快取執行的引擎。接著會出現執行該引擎的所有快取清單。
3. 若要選擇要刪除的快取，請從快取清單中選擇快取的名稱。

#### Important

您一次只能從 ElastiCache 主控台刪除一個快取。選擇多個快取會停用刪除操作。

4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 在刪除快取確認畫面中，選擇刪除以刪除快取，或選擇取消以保留叢集。
6. 如果您選擇刪除，快取的狀態就會變更為正在刪除。

一旦您的快取進入刪除中狀態，就不會再對您收費。

### 使用 AWS CLI

下列程式碼會刪除 my-cache 快取。

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

delete-serverless-cache CLI 動作只會刪除一個無伺服器快取。若要刪除多個快取，請針對您要刪除的每個無伺服器快取呼叫 delete-serverless-cache。您不需要等待某個無伺服器快取完成刪除，即可刪除其他快取。

若為 Linux、macOS 或 Unix：

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

```
--serverless-cache-name my-cache
```

用於 Windows :

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

如需詳細資訊，請參閱 ElastiCache 的 AWS CLI 主題 [delete-serverless-cache](#)。

現在您可以繼續進行 [後續步驟](#)。

## 後續步驟

如需有關 ElastiCache 的詳細資訊，請參閱：

- [使用 ElastiCache](#)
- [縮放記憶 ElastiCache 體快取](#)
- [ElastiCache 的配額](#)
- [ElastiCache 最佳實踐和緩存策略](#)
- [檢視 ElastiCache 事件](#)

# 教學課程：設定 Lambda 函數以在 Amazon VPC 中存取 Amazon ElastiCache

在此教學課程中，您將執行下列操作：

- 在 us-east-1 區域中，於您的預設 Amazon Virtual Private Cloud (Amazon VPC) 中建立 Amazon ElastiCache 快取。
- 建立 Lambda 函數來存取 ElastiCache 快取。在您建立 Lambda 函數時，可在 Amazon VPC 和 VPC 安全群組中提供子網路 ID，以允許 Lambda 函數存取 VPC 中的資源。為了在本教學課程中進行說明，Lambda 函數產生 UUID，將其寫入快取，並從快取中取回。
- 手動調用 Lambda 函數，並驗證它是否已存取您 VPC 中的 ElastiCache 快取。

## Important

本教學課程使用您帳戶 us-east-1 區域中的預設 Amazon VPC。如需 Amazon VPC 的詳細資訊，請參閱《Amazon VPC 使用者指南中的[如何開始 Amazon VPC](#)。

## 主題

- [步驟 1：建立 ElastiCache 快取](#)
- [步驟 2：建立 Lambda 函數](#)
- [步驟 3：測試 Lambda 函數](#)

## 開始使用

### [步驟 1：建立 ElastiCache 快取](#)

## 步驟 1：建立 ElastiCache 快取

在此步驟中，您可以使用 AWS CLI，在帳戶 us-east-1 區域的預設 Amazon Virtual Private Cloud 中建立 Amazon ElastiCache 快取。如需使用 ElastiCache 主控台或 API 建立 ElastiCache 無伺服器快取的相關資訊，請參閱《ElastiCache for Memcached 使用者指南》中的[建立叢集](#)。

## AWS Management Console

執行以下 AWS CLI 命令，在 us-east-1 區域的預設 VPC 中建立 Memcached 叢集無伺服器快取。

## Linux

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

## Windows

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

請注意，「狀態」欄位的值會設定為 CREATING。ElastiCache 需要幾分鐘的時間才能完成建立叢集。

若要確認 ElastiCache 已完成建立快取，請使用 `describe-serverless-caches` 命令。

## Linux

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

## Windows

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

複製輸出中所顯示的端點地址。您為 Lambda 函數建立部署套件時，會需要這個地址。

建立新快取後，繼續進行 [步驟 2：建立 Lambda 函數](#)

Next Step: (後續步驟：)

[步驟 2：建立 Lambda 函數](#)

## 步驟 2：建立 Lambda 函數

請於本步驟執行以下操作：

1. 使用提供的範本程式碼建立 Lambda 函數部署套件。
2. 建立 IAM 角色 (執行角色)。在上傳部署套件時，您需要指定此角色，以便 Lambda 可以擔任此角色來代表您執行此函式。許可政策授予 AWS Lambda 許可權以設定彈性網路界面 (ENI)，使您的 Lambda 函數能夠存取 VPC 中的資源。在本例中，您的 Lambda 函數存取 VPC 中的 ElastiCache 叢集。
3. 透過上傳部署套件來建立 Lambda 函數。

## 後續步驟

### [步驟 2.1：建立部署套件](#)

#### 步驟 2.1：建立部署套件

目前，Lambda 函數的範例程式碼僅以 Python 提供。

#### Python

下面的範例 Python 程式碼將項目讀取並寫入到您的 ElastiCache 叢集。複製程式碼並將它儲存到名為 `app.py` 的檔案中。請務必將程式碼中的 `elasticache_config_endpoint` 值取代為您步驟 1 中複製的端點地址。

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

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

context = ssl.create_default_context()

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

def lambda_handler(event, context):

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

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

```
# get the item (UUID) from the cache
result = memcached_client.get("uuid")
decoded_result = result.decode("utf-8")

# check the retrieved item matches the item added to the cache and print
# the results
if decoded_result == uuid_in:
    print(f"Success: Inserted {uuid_in}. Fetched {decoded_result} from Memcached.")
else:
    raise Exception(f"Bad value retrieved. Expected {uuid_in}, got
{decoded_result}")

return "Fetched value from Memcached"
```

此程式碼會使用 Python [pymemcache](#) 程式庫將項目放入您的快取中，並進行擷取。若要建立包含 pymemcache 的部署套件，請執行下列步驟。

1. 在包含 app.py 原始程式碼檔案的專案目錄中，建立資料夾 package，以將 pymemacache 程式庫安裝到其中。

```
mkdir package
```

2. 使用 pip 安裝 pymemcache。

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

3. 建立包含 pymemcache 程式庫的 .zip 檔案。在 Linux 和 macOS 中，執行下列命令。在 Windows 中，使用您慣用的壓縮公用程式建立 .zip 檔案，並將 pymemache 程式庫放在根目錄。

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

4. 將您的函數程式碼新增至 .zip 檔案。在 Linux 和 macOS 中，執行下列命令。在 Windows 中，使用您慣用的壓縮公用程式將 app.py 新增至 .zip 檔案的根目錄。

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

## 後續步驟

## [步驟 2.2 : 建立 IAM 角色 \(執行角色\)](#)

### 步驟 2.2 : 建立 IAM 角色 (執行角色)

在此步驟中，使用以下預先定義的角色類型和存取政策，建立 AWS 身分和存取管理 (IAM) 角色：

- AWS Lambda 類型的 AWS 服務角色 – 此角色授予 AWS Lambda 擔任此角色的許可。
- AWSLambdaVPCLambdaAccessExecutionRole – 此為您連接到角色的存取許可政策。該政策授予許可給其 ENI 需被 AWS Lambda 所管理的 EC2 動作。您可以在 IAM 主控台檢視此 AWS 受管政策。

如需 IAM 使用者角色的詳細資訊，請參閱《IAM 使用者指南》中的[角色 \(委派與聯合\)](#)。

請使用下列步驟建立 IAM 角色。

#### 建立 IAM (執行) 角色

1. 登入 AWS 管理主控台，然後前往 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。
2. 選擇 Roles (角色)，然後選擇 Create role (建立角色)。
  - 在信任的實體類型下，選擇 AWS 服務，然後在使用案例下選擇 Lambda。這將授與 AWS Lambda 擔任該角色的服務許可。選擇 Next (下一步)。
  - 在新增許可下，搜尋 **AWSLambdaVPCLambdaAccessExecutionRole** 並選取政策旁的核取方塊。
  - 選擇 Next (下一步)。
  - 在 Role Name (角色名稱) 中，使用您 AWS 帳戶中的唯一名稱 (例如，lambda-vpc-execution-role)。
  - 選擇建立角色。
3. 複製角色 ARN。在下一個步驟建立您的 Lambda 函數時會需要用到它。

#### 後續步驟

## [步驟 2.3 : 上傳部署套件 \(建立 Lambda 函數\)](#)

### 步驟 2.3 : 上傳部署套件 (建立 Lambda 函數)

在此步驟中，您會使用 create-function AWS CLI 命令建立 Lambda 函數 (AccessMemcached)。

從包含部署套件 .zip 檔案的專案目錄，執行下列 Lambda CLI create-function 命令。

針對 `role` 選項，使用您在步驟 2.2 中建立的執行角色的 ARN。針對 `vpc-config`，輸入預設 VPC 子網路及預設 VPC 的安全群組 ID 的逗號分隔清單。您可以在 [Amazon VPC 主控台](#) 中找到這些值。若要尋找預設 VPC 的子網路，請選擇您的 VPC，然後選擇您的 AWS 帳戶的預設 VPC。若要尋找此 VPC 的安全群組，請在安全性下選擇安全群組。確認已選取 `us-east-1` 區域。

若為 Linux、macOS 或 Unix：

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

```

針對 Windows：

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

```

您也可以選擇上傳 zip 檔至相同 AWS 區域中的 Amazon S3 儲存貯體，然後以前述命令指定該儲存貯體與物件名稱。您需要以 `--code` 參數取代 `--zip-file` 參數，如下所示：

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

您也可以使用 AWS Lambda 主控台建立 Lambda 函數。建立函式時，請為 Lambda 選擇一個 VPC，然後從提供的欄位中選擇子網路和安全群組。

後續步驟

## 步驟 3：測試 Lambda 函數

### 步驟 3：測試 Lambda 函數

在這一步，使用 `invoke` 命令手動調用 Lambda 函數。當 Lambda 函數執行時，它會產生 UUID 並將其寫入您在 Lambda 程式碼中指定的 ElastiCache 叢集。然後 Lambda 函數從快取中取回項目。

1. 使用 AWS Lambda `invoke` 命令叫用 Lambda 函數 (AccessMemCache)。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

2. 確認 Lambda 函數是否成功執行，如下：

- 檢視 `output.txt` 檔案。
- 開啟 [CloudWatch](#) 主控台並選擇您函數 (`/aws/lambda/AccessMemcached`) 的日誌群組，以在 CloudWatch Logs 中驗證結果。日誌串流應包含類似以下的輸出內容：

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

- 在 AWS Lambda 主控台檢視結果。

# ElastiCache 教學和影片

以下教學著重在 Amazon ElastiCache 使用者感興趣的任務。

- [ElastiCache 影片](#)
- [教學：設定 Lambda 函式以在 Amazon VPC 中存取 Amazon ElastiCache](#)

## ElastiCache 影片

下方為可協助您了解 Amazon ElastiCache 基本及進階概念的影片。如需 AWS 培訓的詳細資訊，請參閱 [AWS 培訓與認證](#)。

### 主題

- [介紹影片](#)
- [進階影片](#)

### 介紹影片

以下影片向您介紹 Amazon ElastiCache。

### 主題

- [AWS re:Invent 2020 年會：Amazon ElastiCache 的新功能](#)
- [AWS re:Invent 2019 年會：Amazon ElastiCache 的新功能](#)
- [AWS re:Invent 2017 年會：Amazon ElastiCache 的新功能](#)
- [Data204 - 在 AWS NoSQL 服務上建置可擴展的應用程式 \(re:Invent 2015 年會\)](#)
- [DAT207 - 使用 Amazon ElastiCache 加速應用程式效能 \(AWS re:Invent 2013 年會\)](#)

[AWS re:Invent 2020 年會：Amazon ElastiCache 的新功能](#)

[AWS re:Invent 2020 年會：Amazon ElastiCache 的新功能](#)

[AWS re:Invent 2019 年會：Amazon ElastiCache 的新功能](#)

[AWS re:Invent 2019 年會：Amazon ElastiCache 的新功能](#)

[AWS re:Invent 2017 年會：Amazon ElastiCache 的新功能](#)

[AWS re:Invent 2017 年會：Amazon ElastiCache 的新功能](#)

[Data204 - 在 AWS NoSQL 服務上建置可擴展的應用程式 \(re:Invent 2015 年會\)](#)

在本課程中，我們會討論 NoSQL 資料庫的優點，並探索 AWS 提供的主要 NoSQL 服務：Amazon DynamoDB 和 Amazon ElastiCache。然後，我們會聆聽兩位居領導地位的客戶，Expedia 和 Mapbox，了解他們的使用案例和架構挑戰，以及他們如何使用 AWS NoSQL 服務面對挑戰，包括設計模式和最佳實務。建議您收看此研討會來更進一步了解 NoSQL 及其強大的功能，並準備好充滿信心的面對資料庫的挑戰。

## [DAT204 - 在 AWS NoSQL 服務上建置可擴展的應用程式 \(re:Invent 2015 年會\)](#)

## [DAT207 - 使用 Amazon ElastiCache 加速應用程式效能 \(AWS re:Invent 2013 年會\)](#)

在此影片中，了解如何使用 Amazon ElastiCache 輕鬆部署記憶體內快取系統來加速您的應用程式效能。我們會向您示範如何使用 Amazon ElastiCache 改善您應用程式的延遲，並減少資料庫伺服器上的負載。我們也會向您示範如何建置可輕鬆管理，並且可隨著您應用程式成長而擴展的快取 layer。在本課程中，我們會探討各種案例及可藉由啟用快取而受惠的使用案例，並討論 Amazon ElastiCache 提供的功能。

## [DAT207 - 使用 Amazon ElastiCache 加速應用程式效能 \(re:Invent 2013 年會\)](#)

### 進階影片

以下影片涵蓋更進階的 Amazon ElastiCache 主題。

#### 主題

- [使用 Amazon ElastiCache 最佳實務打造邁向成功的設計 \(re:Invent 2020 年會\)](#)
- [使用 Amazon ElastiCache 增強您的即時應用程式 \(re:Invent 2019 年會\)](#)
- [最佳實務：將 Redis 叢集從 Amazon EC2 遷移到 ElastiCache \(re:Invent 2019 年會\)](#)
- [使用 Amazon ElastiCache 和 Amazon Aurora STP11 擴展 Fantasy Sports Platform \(re:Invent 2018 年會\)](#)
- [透過 Amazon ElastiCache 在雲端中使用可靠且可擴展的 Redis \(re:Invent 2018 年會\)](#)
- [深入探索 ElastiCache：記憶體內資料儲存的设计模式 \(Re: Invent 2018 年會\)](#)
- [DAT305 - 深入探討 Amazon ElastiCache \(re:Invent 2017 年會\)](#)
- [DAT306 - 深入探討 Amazon ElastiCache \(re:Invent 2016 年會\)](#)
- [DAT407 - 深入探討 Amazon ElastiCache \(re:Invent 2015 年會\)](#)
- [DAT402 - 深入探討 Amazon ElastiCache \(re:Invent 2014 年會\)](#)
- [DAT307 - 深入探討 Amazon ElastiCache 架構與設計模式 \(re:Invent 2013 年會\)](#)

## [使用 Amazon ElastiCache 最佳實務打造邁向成功的設計 \(re:Invent 2020 年會\)](#)

隨著 Redis 建置於的關鍵業務即時應用程式呈爆炸性成長，可用性、可擴展性和安全性儼然成為首要考量因素。了解設定 Amazon ElastiCache 的最佳實務，透過線上擴展、跨異地同步備份部署的高可用性以及安全組態獲得成功。

## [使用 Amazon ElastiCache 最佳實務打造邁向成功的設計 \(re:Invent 2020 年會\)](#)

## 使用 Amazon ElastiCache 增強您的即時應用程式 (re:Invent 2019 年會)

隨著雲端採用的快速成長以及這股趨勢帶來的新使用情境，應用程式需要微秒延遲和高輸送量，才能支援每秒數百萬個請求。開發人員傳統上依賴專門的硬體和因應措施 (例如磁碟資料庫結合資料縮減技術)，藉此管理即時應用程式的資料。這些做法可能很所費不貲，而且不可擴展。了解如何使用全受管記憶體內 Amazon ElastiCache，獲得極致的效能、高可擴展性、可用性和安全性，進而提高即時應用程式的效能。

### [使用 Amazon ElastiCache 增強您的即時應用程式 \(re:Invent 2019 年會\)](#)

#### 最佳實務：將 Redis 叢集從 Amazon EC2 遷移到 ElastiCache (re:Invent 2019 年會)

由您自行管理 Redis 叢集可能很困難。您必須時常佈建硬體、修補軟體、備份資料以及監控工作負載。運用新發行的 Amazon ElastiCache 線上遷移功能，您現在可以輕鬆將資料從 Amazon EC2 上的自我託管 Redis 移至全受管的 Amazon ElastiCache，並停用叢集模式。在本課程中，您將認識新的線上遷移工具、觀看示範，更重要的是學習順暢遷移到 Amazon ElastiCache 的實際操作最佳實務。

### [最佳實務：將 Redis 叢集從 Amazon EC2 遷移到 ElastiCache \(re:Invent 2019 年會\)](#)

## 使用 Amazon ElastiCache 和 Amazon Aurora STP11 擴展 Fantasy Sports Platform (re:Invent 2018 年會)

Dream11 是印度領先業界的體育科技新創公司。他們超過 4 千萬的成長中使用者群涉及各種運動，包括 Fantasy 板球、足球和籃球，目前可同時為 100 萬名使用者提供服務，且能在 50 毫秒回應時間內每分鐘產生 300 萬個請求。在這場演講中，Dream11 CTO Amit Sharma 說明了該公司如何使用 Amazon Aurora 和 Amazon ElastiCache 來處理瞬間流量，這類流量可能在 30 秒的回應時間內暴增三倍。Sharma 也會討論如何在不進行鎖定的情況下擴展交易量，並分享處理瞬間流量的步驟，進而為 500 萬使日活躍使用者提供服務。完整標題：AWS re:Invent 2018 年會：使用 Amazon ElastiCache 和 Amazon Aurora (STP11) 擴展 Fantasy Sports Platform

### [使用 Amazon ElastiCache 和 Amazon Aurora STP11 擴展 Fantasy Sports Platform \(re:Invent 2018 年會\)](#)

#### 透過 Amazon ElastiCache 在雲端中使用可靠且可擴展的 Redis (re:Invent 2018 年會)

本課程涵蓋 Redis 相容服務 (Amazon ElastiCache for Redis) 的功能和增強功能。我們會說明重要功能 (例如 Redis 5)、可擴展性和效能改善項目、安全性和合規性等等。我們也會討論即將推出的功能和客戶使用案例。

### [透過 Amazon ElastiCache 在雲端中使用可靠且可擴展的 Redis \(re:Invent 2018 年會\)](#)

## 深入探索 ElastiCache：記憶體內資料儲存的设计模式 (Re: Invent 2018 年會)

在本課程中，我們會一窺幕後真相，了解 Amazon ElastiCache 的设计和架構。了解 Redis 與 Memcached 服務項目的共同设计模式，以及客戶如何將這些應用於記憶體內資料處理，以減少延遲並改善應用程式輸送量。我們會介紹 ElastiCache 最佳實務、设计模式及不適合的使用模式。

### [深入探索 ElastiCache：記憶體內資料儲存的设计模式 \(Re: Invent 2018 年會\)](#)

## DAT305 - 深入探討 Amazon ElastiCache (re:Invent 2017 年會)

一窺幕後真相，了解 Amazon ElastiCache 的设计和架構。了解我們 Memcached 與 Redis 服務項目的共同设计模式，以及客戶如何將這些應用於記憶體內操作，以減少延遲並改善應用程式輸送量。影片中會介紹 ElastiCache 最佳實務、设计模式及不適合的使用模式。

影片會介紹以下內容：

- ElastiCache for Redis 線上重新分片
- ElastiCache 安全性與加密
- ElastiCache for Redis 3.2.10 版

### [DAT305 - 深入探討 Amazon ElastiCache \(re:Invent 2017 年會\)](#)

## DAT306 - 深入探討 Amazon ElastiCache (re:Invent 2016 年會)

一窺幕後真相，了解 Amazon ElastiCache 的设计和架構。了解我們 Memcached 與 Redis 服務項目的共同设计模式，以及客戶如何將這些應用於記憶體內操作，以減少延遲並改善應用程式輸送量。在本課程中，我們會說明 ElastiCache 的最佳實務、设计模式及不適合的使用模式。

### [DAT306 - 深入探討 Amazon ElastiCache \(re:Invent 2016 年會\)](#)

## DAT407 - 深入探討 Amazon ElastiCache (re:Invent 2015 年會)

一窺幕後真相，了解 Amazon ElastiCache 的设计和架構。了解我們 Memcached 與 Redis 服務項目的共同设计模式，以及客戶如何將這些應用於記憶體內操作，以改善延遲和應用程式的輸送量。於本課程中，我們會介紹與 Amazon ElastiCache 有關的最佳實務、设计模式及不適合的使用模式。

### [DAT407 - 深入探討 Amazon ElastiCache \(re:Invent 2015 年會\)](#)

## DAT402 - 深入探討 Amazon ElastiCache (re:Invent 2014 年會)

在此影片中，我們會檢視常見的快取使用案例、Memcached 與 Redis 引擎、有助您判斷哪個引擎更符合您需求的模式、一致性雜湊，以及更多可協助您建置快速、可擴展應用程式的工具。Adobe 的主任

研究員 Frank Wiebe 會詳細說明 Adobe 如何使用 Amazon ElastiCache 改善客戶體驗及擴展他們的事業。

[DAT402 - 深入探討 Amazon ElastiCache \(re:Invent 2014 年會\)](#)

DAT307 - 深入探討 Amazon ElastiCache 架構與設計模式 (re:Invent 2013 年會)

在此影片中，我們會檢視快取、策略、向外擴展與監控。我們也會比較 Memcached 與 Redis 引擎。於本課程中，我們也會介紹與 Amazon ElastiCache 有關的最佳實務和設計模式。

[DAT307 - 深入探討 Amazon ElastiCache 架構與設計模式 \(AWS re:Invent 2013 年會\)](#)。

## 選擇區域與可用區域

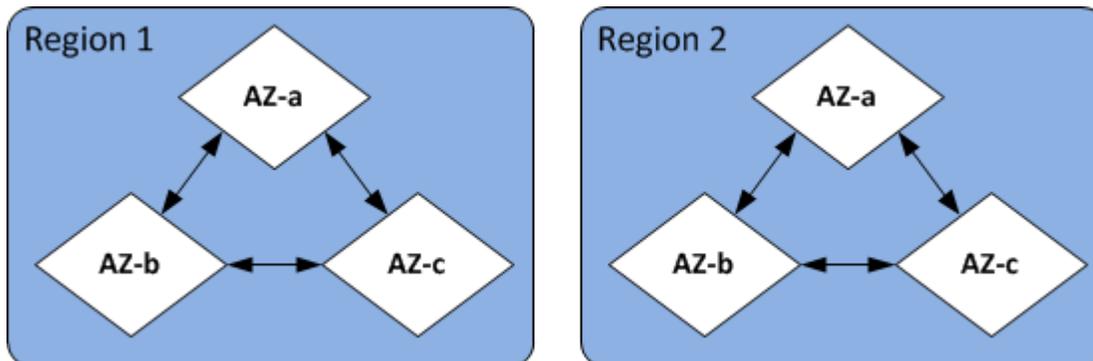
AWS 雲計算資源存放在高可用性的數據中心設施中。為了提供額外的可擴展性和可靠性，這些資料中心設施會位在不同的實體位置。這些地點是依「區域」及「可用區域」分類。

AWS 區域規模龐大，廣泛分散到單獨的地理位置。可用區域是區域內的不同位置，其設計目的是與其他可用區域中的故障隔離。AWS 它們為相 AWS 同區域中的其他可用區域提供廉價、低延遲的網路連線能力。

### ⚠ Important

每個區域都是完全獨立的。您啟動的任何 ElastiCache 活動 (例如，建立叢集) 只會在目前的預設區域中執行。

若要在特定區域中建立或使用叢集，請使用對應的區域服務端點。如需了解服務端點，請參閱[支援的區域與端點](#)。



### 區域與可用區域

#### 主題

- [可用區域的考量](#)
- [支援的區域與端點](#)
- [安置您的節點](#)
- [搭配 ElastiCache 使用本地區域](#)
- [使用 Outposts](#)

## 可用區域的考量

將 Memcached 節點散佈在一個區域內的多個可用區域，有助於保護您免受災難性故障的影響，例如可用區域內停電。

### 無伺服器快取

ElastiCache 無伺服器快取會建立跨越多個可用區域的高可用性快取。您可以在建立快取時指定來自不同可用區域和相同 VPC 的子網路，或 ElastiCache 從預設 VPC 自動選擇子網路。

ElastiCache 為內存緩存集群設計自己的

Memcached 叢集最多可以有 300 個節點。當您建立或新增節點至 Memcached 叢集時，您可以為所有節點指定單一可用區域、允許 ElastiCache 為所有節點選擇單一可用區域、為每個節點指定可用區域，或允許 ElastiCache 為每個節點選擇可用區域。將新節點新增至現有的 Memcached 叢集時，您可以在不同的可用區域中建立新節點。快取節點一旦建立，就無法修改其可用區域。

如果您希望單一可用區域叢集中的叢集使其節點分散到多個可用區域，ElastiCache 可以在各個可用區域中建立新節點。接著您可以刪除部分或所有原始快取節點。建議您使用此方法。

將 Memcached 節點從單一可用區域遷移至多個可用區域

1. 在需要的可用區域中建立新的快取節點，藉此修改叢集。在請求中執行下列操作：
  - 將 AZMode (CLI : `--az-mode`) 設定為 `cross-az`。
  - 將 NumCacheNodes (CLI : `--num-cache-nodes`) 設定為目前作用中快取節點的數目，以及您要建立的新快取節點數目。
  - 將 NewAvailabilityZones (CLI : `--new-availability-zones`) 設定為您要在其中建立新快取節點的區域清單。若要讓 ElastiCache 決定每個新節點的可用區域，請勿指定清單。
  - 將 ApplyImmediately (CLI : `--apply-immediately`) 設定為 `true`。

#### Note

如果您沒有使用自動探索功能，請務必使用新的快取節點端點來更新用戶端應用程式。

繼續進行下一個步驟之前，請確定 Memcached 節點已完全建立且可用。

2. 移除原始可用區域中不再需要的節點，藉此修改叢集。在請求中執行下列操作：

- 將 NumCacheNodes (CLI : --num-cache-nodes) 設定為套用此修改之後您想要的作用中快取節點數目。
- 將 CacheNodeIdsToRemove (CLI : --nodes-to-remove) 設定為您要從叢集中移除的快取節點清單。

列出的快取節點 ID 數量必須等於目前作用中的節點數減去 NumCacheNodes 中的值。

- (選用) 將 ApplyImmediately (CLI : --apply-immediately) 設定為 true。

如果您沒有將 ApplyImmediately (CLI : --apply-immediately) 設定為 true , 節點刪除作業將在下一個維護時段進行。

## 支援的區域與端點

Amazon 可 ElastiCache 在多個 AWS 區域使用。這表示您可以在符合需求的位置啟動 ElastiCache 叢集。例如，您可以在離客戶最近的 AWS 地區啟動，或在特定 AWS 區域啟動以符合特定法律要求。

每個區域皆設計為與其他區域完全隔離。各個區域包含多個可用區域 (AZ)。ElastiCache 無伺服器快取會自動跨多個可用區域複寫資料 (除了在兩個可用區域中複寫資料除外)us-west-1，以提供高可用性。在設計自己的 ElastiCache 叢集時，您可以選擇在不同的 AZ 中啟動節點，以達到容錯能力。如需區域和可用區域的詳細資訊，請參閱本主題最上方的 [選擇區域與可用區域](#)。

### 支援 ElastiCache 的地區

區域名稱/區域	端點	通訊協定
美國東部 (俄亥俄) 區域 us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS
美國東部 (維吉尼亞北部) 區域 us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS
美國西部 (加利佛尼亞北部) 區域 us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS
美國西部 (奧勒岡) 區域 us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS
加拿大 (中部) 區域 ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS

區域名稱/區域	端點	通訊協定
加拿大 (西部) 區域 ca-west-1	elasticache.ca-west-1.amazonaws.com	HTTPS
亞太區域 (雅加達) ap-southeast-3	elasticache.ap-southeast-3.amazonaws.com	HTTPS
亞太 (孟買) 區域 ap-south-1	elasticache.ap-south-1.amazonaws.com	HTTPS
亞太區域 (海德拉巴) ap-south-2	elasticache.ap-south-2.amazonaws.com	HTTPS
亞太 (東京) 區域 ap-northeast-1	elasticache.ap-northeast-1.amazonaws.com	HTTPS
亞太 (首爾) 區域 ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS
亞太 (大阪) 區域 ap-northeast-3	elasticache.ap-northeast-3.amazonaws.com	HTTPS
亞太 (新加坡) 區域 ap-southeast-1	elasticache.ap-southeast-1.amazonaws.com	HTTPS
亞太 (雪梨) 區域 ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS

區域名稱/區域	端點	通訊協定
歐洲 (法蘭克福) 區域 eu-central-1	elasticache.eu- central-1.am azonaws.com	HTTPS
歐洲 (蘇黎世) 區域 eu-central-2	elasticache.eu- central-2.am azonaws.com	HTTPS
歐洲 (斯德哥爾摩) 區域 eu-north-1	elasticache.eu- north-1.amaz onaws.com	HTTPS
Middle East (Bahrain) Region me-south-1	elasticache.me- south-1.amaz onaws.com	HTTPS
中東 (阿拉伯聯合大公 國) 區域 me-central-1	elasticache.me- central-1.am azonaws.com	HTTPS
歐洲 (愛爾蘭) 區域 eu-west-1	elasticache.eu- west-1.amazo naws.com	HTTPS
歐洲 (倫敦) 區域 eu-west-2	elasticache.eu- west-2.amazo naws.com	HTTPS
歐洲 (巴黎) 區域 eu-west-3	elasticache.eu- west-3.amazo naws.com	HTTPS

區域名稱/區域	端點	通訊協定
Europe (Milan) Region eu-south-1	elasticache.eu-south-1.amazonaws.com	HTTPS
歐洲 (西班牙) 區域 eu-south-2	elasticache.eu-south-2.amazonaws.com	HTTPS
南美洲 (聖保羅) 區域 sa-east-1	elasticache.sa-east-1.amazonaws.com	HTTPS
中國 (北京) 區域 cn-north-1	elasticache.cn-north-1.amazonaws.com.cn	HTTPS
中國 (寧夏) 區域 cn-northwest-1	elasticache.cn-northwest-1.amazonaws.com.cn	HTTPS
亞太區域 (香港) 區域 ap-east-1	elasticache.ap-east-1.amazonaws.com	HTTPS
非洲 (開普敦) 區域 af-south-1	elasticache.af-south-1.amazonaws.com	HTTPS
以色列 (特拉維夫) 區域 il-central-1	elasticache.il-central-1.amazonaws.com	HTTPS

區域名稱/區域	端點	通訊協定
AWS GovCloud (美國西部) us-gov-west-1	elasticache.us-gov-west-1.amazonaws.com	HTTPS
AWS GovCloud (美國東部) us-gov-east-1	elasticache.us-gov-east-1.amazonaws.com	HTTPS

如需搭配使用 AWS GovCloud (US) 的相關資訊 ElastiCache，請參閱 [AWS GovCloud \(美國\) 地區的服務：ElastiCache](#)。

有些區域支援一部分的節點類型。如需「AWS 區域」支援節點類型的表格，請參閱 [支援的節點類型 \(依 AWS 區域\)](#)。

如需按地區分 AWS 類的產品與服務表格，請參閱按地 [區分類的產品與服務](#)。

## 安置您的節點

Amazon ElastiCache 支援在單一或多個可用區域 (AZ) 中定位叢集的所有節點。此外，如果您選擇在多個 AZ 中尋找節點 (建議使用)，ElastiCache 可讓您為每個節點選擇 AZ，或允許 ElastiCache 為您選擇節點。

將節點安置在不同 AZ 時，您可以避免某個 AZ 的故障 (例如停電) 就造成整個系統故障。

您可以在建立叢集時為每個節點指定 AZ，或在修改現有叢集時新增節點來完成此作業。如需詳細資訊，請參閱下列內容：

- [建立叢集](#)
- [修改 ElastiCache 叢集](#)
- [將節點新增至叢集](#)

## 搭配 ElastiCache 使用本地區域

本機區域是 AWS 區域與使用者地理位置相近的擴充功能。您可以藉由建立新的子網路並指派至本地區域，將上層 AWS 區域的任何 Virtual Private Cloud (VPC) 延伸至本地區域。當您在本機區域中建立子網路時，VPC 也會擴展到該本機區域。本機區域中的子網路與 VPC 中的其他子網路的運作方式相同。

使用本地區域，您可將資源 (例如 ElastiCache 叢集) 放置在靠近使用者的多個位置。

建立 ElastiCache 叢集時，您可以選擇本地區域的子網路。Local Zones 有自己的網際網路連線，並支援 AWS Direct Connect。因此，在本機區域中建立的資源可以為本機使用者提供非常低延遲的通訊服務。如需詳細資訊，請參閱 [AWS Local Zones](#)。

本地區域是以一個 AWS 區域代碼加上一個識別符來表示，用以代表位置，例如 us-west-2-lax-1a。

目前可用的本地區域有 us-west-2-lax-1a 和 us-west-2-lax-1b。

ElastiCache 適用下列本地區域限制：

- 本地區域目前支援以下節點類型：
  - 最新一代：

M5 節點類

型：cache.m5.large、cache.m5.xlarge、cache.m5.2xlarge、cache.m5.4xlarge、cache.

## R5 節點類

型：`cache.r5.large`、`cache.r5.xlarge`、`cache.r5.2xlarge`、`cache.r5.4xlarge`、`cache.`

T3 節點類型：`cache.t3.micro`、`cache.t3.small`、`cache.t3.medium`

## 啟用本地區域

1. 在 Amazon EC2 主控台中啟用本機區域。

如需詳細資訊，請參閱 Amazon EC2 使用者指南中的[啟用本地區域](#)。

2. 在本機區域中建立子網路。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[在您的 VPC 中建立子網路](#)。

3. 在本地區域中建立 ElastiCache 子網路群組。

建立 ElastiCache 子網路群組時，請選擇本地區域的可用區域群組。

如需詳細資訊，請參閱 ElastiCache 使用者指南中的[建立子網路群組](#)。

4. 在本地區域中建立使用 ElastiCache 子網路的 ElastiCache for Memcached 叢集。

如需更多詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。

## 使用 Outposts

AWS Outposts 是一項全受管服務，可將 AWS 基礎架構、服務、API 和工具擴展到客戶場所。透過提供 AWS 受管理基礎架構的本機存取權，AWS Outposts 可讓客戶使用與 AWS 區域相同的程式設計介面在內部部署建置和執行應用程式，同時使用本機運算和儲存資源來降低延遲和本機資料處理需求。Outpost 是部署在客戶站點的 AWS 計算和儲存容量集區。AWS 作為 AWS 區域的一部分來操作、監控和管理此容量。您可以在 Outpost 上建立子網路，並在建立叢集等 AWS 資源時指定子網路。

ElastiCache

### Note

此版本具有下列限制：

- ElastiCache 對於 Outposts 僅支持 M5 和 R5 節點家族。
- 多可用區域 (不支援跨 Outpost 複寫)。

- ElastiCache 在 Outposts 不支持 CoIP。
- ElastiCache 對於 Outposts 在以下地區不支持：CN-北 -1，CN-西北 -1 和 ap-東北 -3。

## 透過 Memcached 主控台使用 Outpost

1. 請登入 AWS Management Console 並開啟 ElastiCache 主控台，網址為 <https://console.aws.amazon.com/elasticache/>。
2. 在導航窗格中，選擇內存緩存。
3. 選擇創建內存緩存。
4. 在 [叢集設定] 下，選取 [設計您自己的快取和叢集快取]。保持叢集模式設定為 [停用]。然後為緩存創建一個名稱和可選描述。
5. 對於位置，請選擇在內部部署。
6. 在「內部部署」部分中，您將看到該字段前哨 ID。輸入叢集執行位置的 ID。

叢集設定下的所有進一步設定都可以保留為預設值。

7. 在 [連線] 中，選取 [建立新的子網路群組]，然後輸入 VPC ID。將其餘部分保留為預設值，然後選取下一步。

### 設定內部部署選項

您可以選取可用的 Outpost 來新增快取叢集，或者如果沒有可用的 Outpost，請使用下列步驟建立新的 Outpost：

在 On-Premises options (內部部署選項) 底下：

1. 在 Memcached settings (Memcached 設定) 底下：
  - a. Name (名稱)：輸入 Memcached 叢集的名稱
  - b. Description (描述)：輸入 Memcached 叢集的描述。
  - c. 引擎版本相容性：引擎版本以 AWS 前哨區域為基礎
  - d. Port (連接埠)：接受預設連接埠 11211。如果您要使用不同的連接埠，請輸入連接埠號碼。
  - e. Parameter group (參數群組)：使用下拉式選單選取預設或自訂參數群組。
  - f. Node Type (節點類型)：可用的執行個體取決於 Outpost 可用性。從下拉式清單中，選取 Outposts，然後選取您要用於此叢集的可用節點類型。然後選取 Save (儲存)。

- g. Number of nodes (節點數量)：輸入您希望叢集包含的節點數量。
2. 在連線下：
    - a. Subnet Group (子網路群組)：從清單中選取 Create new (新建)。
      - Name (名稱)：輸入子網路群組的名稱
      - Description (描述)：輸入子網路群組的描述
      - VPC ID：VPC ID 應與 Outpost VPC 一致。
      - Availability Zone or Outpost (可用區域或 Outpost)：選取您正在使用的 Outpost。
      - Subnet ID (子網路 ID)：選取可用於 Outpost 的子網路 ID。如果沒有可用的子網路 ID，您需要建立。如需詳細資訊，請參閱「[建立子網路](#)」。
    - b. 選取建立。

## 檢視 Outpost 叢集詳細資訊

在 Memcached 清單頁面上，選取屬於 AWS Outpost 的叢集，並在檢視叢集詳細資料時注意下列事項：

- 可用區域：這將代表前哨，使用 ARN (Amazon 資源名稱) 和資 AWS 源編號。
- 前哨名稱：前 AWS 哨的名稱。

## 使用 Outposts 與 CLI AWS

您可以使用 AWS Command Line Interface (AWS CLI) 從命令列控制多個 AWS 服務，並透過指令碼將它們自動化。您可以使用 AWS CLI 進行臨機操作（一次性）操作。

### 下載和配置 AWS CLI

在視窗上 AWS CLI 運行, macOS 系統, 或 Linux. 請使用下列程序來下載並進行設定。

#### 下載、安裝及設定 CLI

1. 在[AWS 命令列介面](#)網頁上下載 AWS CLI。
2. 請遵循《AWS Command Line Interface 使用者指南》[中有關安裝 AWS CLI](#) 和 [設定 AWS CLI](#) 的說明進行操作。

## 使用 AWS CLI 與 Outposts

使用下列 CLI 作業來建立使用 Outpost 的快取叢集：

- [create-cache-cluster](#)— 使用此作業，`outpost-mode` 參數會接受值，此值會指定快取叢集中的節點是在單一 Outpost 中建立，還是跨多個 Outposts。

### Note

目前只支援 `single-outpost` 模式。

```
aws elasticache create-cache-cluster \  
  --cache-cluster-id cache cluster id \  
  --outpost-mode single-outpost \  
  \
```

# 自行設計和管理 Memcached 實作的 ElastiCache 叢集

如果您需要精準控制 ElastiCache 叢集，可以選擇自行設計叢集。ElastiCache 可讓您藉由為叢集選擇節點類型、節點數和 AWS 可用區域內的節點放置，來操作節點型叢集。由於 ElastiCache 是一項全受管服務，因此可自動管理叢集的硬體佈建、監控、節點替換和軟體修補。

如需設定的詳細資訊，請參閱 [設定](#)。如需管理、更新或刪除節點或叢集的詳細資訊，請參閱 [管理節點](#)。如需自行設計 ElastiCache 叢集時，Amazon ElastiCache 部署的主要元件概觀，請參閱這些 [重要概念](#)。

## 主題

- [ElastiCache 用於內存緩存的組件和功能](#)
- [管理叢集](#)
- [管理節點](#)

# ElastiCache 用於內存緩存的組件和功能

接下來，你可以找到一個 Amazon ElastiCache 的 Memcached 部署的主要組件的概述。

## 主題

- [ElastiCache 節點](#)
- [ElastiCache 用於記憶體快取叢集](#)
- [AWS 區域和可用區域](#)
- [ElastiCache 用於記憶體快取端點](#)
- [ElastiCache 參數群組](#)
- [ElastiCache 安全](#)
- [ElastiCache 子網路群組](#)
- [ElastiCache 對於內存緩存事件](#)

## ElastiCache 節點

節點是部署中最小的建 ElastiCache 置區塊。節點可以與其他節點隔離而獨立存在，或與其他節點以某種關係存在。

節點是大小固定、與網路連接的安全 RAM 區塊。每個節點都會執行自己的 Memcached 執行個體。您可視需要將叢集中的節點垂直擴展或縮減為不同的執行個體類型。如需詳細資訊，請參閱 [縮放記憶 ElastiCache 體快取](#)。

叢集中每個節點的執行個體類型都相同，並且執行相同的快取引擎。每個快取節點都有自己的網域名稱服務 (DNS) 名稱和連接埠。支援多種類型的快取節點，每個節點都有各種數量的相關記憶體。如需支援的節點執行個體類型清單，請參閱 [支援的節點類型](#)。

您可以購買節點的 pay-as-you-go 基礎上，您只需支付使用節點的費用。或以大幅降低的每小時費率購買預留節點。如果您的用量很高，購買預留節點可以為您節省成本。假設您的叢集幾乎總是在使用中，且您會偶爾新增節點來處理使用尖峰。在這種情況下，您可以購買多個預留節點來執行大部分時間，並針對偶爾需要新增 pay-as-you-go 節點的時間購買節點。如需預留節點的詳細資訊，請參閱 [ElastiCache 預留節點](#)。

Memcached 引擎支援自動探索。「自動探索」可讓用戶端程式在快取叢集中自動識別所有節點，並初始化及維持與這些節點的連線。透過自動探索，您的應用程式即無需手動連線到各個節點。反之，您的

應用程式會連線到組態端點。組態端點 DNS 項目包含每個快取節點端點的 CNAME 項目。因此，藉由連線至組態端點，您的應用程式會立即取得叢集中所有節點資訊，並可以連線至所有節點。您不需要在應用程式中硬式編碼個別快取節點端點。如需詳細資訊，請參閱[自動探索](#)。

如需節點的詳細資訊，請參閱[管理節點](#)。

## ElastiCache 用於記憶體快取叢集

Memcached「叢集」是一或多個 [ElastiCache 節點](#) 的邏輯分組。資料會分割到 Memcached 叢集的各節點中。

許多 ElastiCache 作業都針對叢集：

- 建立叢集
- 修改叢集
- 刪除叢集
- 在叢集中檢視元素
- 從叢集中新增或移除成本配置標籤

如需詳細資訊，請參閱下列相關主題：

- [管理叢集](#) 和 [管理節點](#)

叢集、節點和相關操作的詳細資訊。

- [AWS 服務限制：Amazon ElastiCache](#)

有關 ElastiCache 限制的資訊，例如節點或叢集的數目上限。

如果您需要超過這些限制，請使用 [Amazon ElastiCache 快取節點請求表提出請求](#)。

- [減少故障](#)

提升叢集容錯能力的相關資訊。

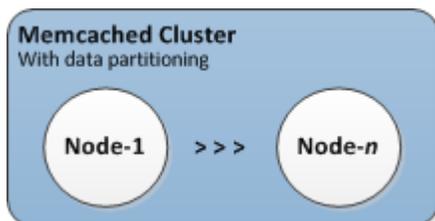
### 一般叢集組態

Memcached 為每個 AWS 區域的每個客戶最多支援 300 個節點，每個叢集具有 1-60 個節點。您可將您的資料分割到 Memcached 叢集的各節點中。

當您執行 Memcached 引擎時，叢集可以由 1-60 個節點組成。您可將您的資料庫分割至各節點中。您的應用程式會讀取和寫入每個節點的端點。如需詳細資訊，請參閱[自動探索](#)。

若要改善容錯能力，請在叢集區域內的各個可用區域 (AZ) 中找出 Memcached 節點。AWS 如此一來，單一 AZ 中的故障，對整個叢集和應用程式的影響會降到最低。如需詳細資訊，請參閱[減少故障](#)。

隨著 Memcached 叢集的需求變更，您可以透過新增或移除節點來擴展或縮減節點，在節點的新數量之間重新分割資料。當您分割資料時，建議您使用一致性雜湊。如需一致性雜湊的詳細資訊，請參閱[設定您的 ElastiCache 用戶端以達到高效率的負載平衡](#)。在下圖中，您可以看到單一節點和多節點 Memcached 叢集的範例。

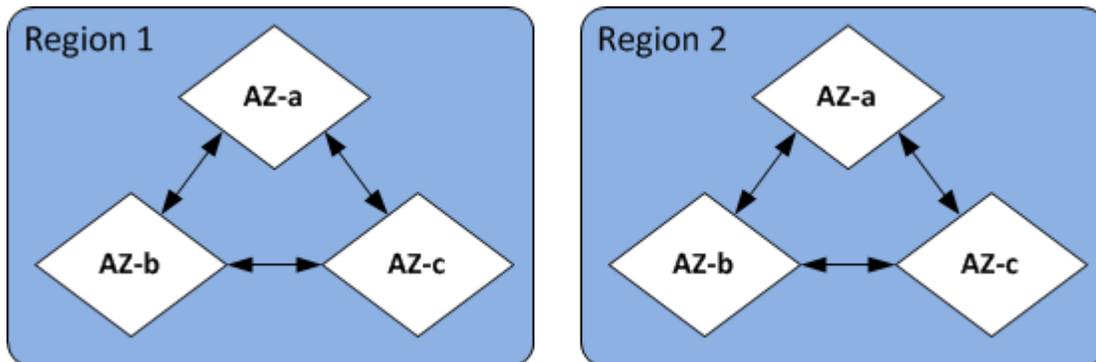


## AWS 區域和可用區域

適用 ElastiCache 於 Memcached 的 Amazon 可在全球多個 AWS 區域使用。因此，您可以在符合業務需求的位置啟動 ElastiCache 叢集。例如，您可以在離您客戶最近的 AWS 地區啟動，或者滿足某些法律要求。

依預設，AWS SDK AWS CLI、ElastiCache API 和 ElastiCache 主控台會參照美國西部 (奧勒岡) 區域。隨著將可用性 ElastiCache 擴展到新 AWS 區域，這些 AWS 區域的新端點也可用於 HTTP 請求 AWS CLI、AWS SDK 和 ElastiCache 主控台。

每個 AWS 區域都被設計為與其他區 AWS 域完全隔離。每個區域中皆有多個可用區域。藉由在不同的可用區域中啟動您的節點，您可以實現最大的容錯能力。如需區域和可用區 AWS 域的相關資訊，請參閱[選擇區域與可用區域](#)。



如需有關受支援的 AWS 區域 ElastiCache 及其端點的資訊，請參閱 [支援的區域與端點](#)。

## ElastiCache 用於記憶體快取端點

端點是應用程式用來連線到 ElastiCache 節點或叢集的唯一位址。

Memcached 叢集中的每個節點都有自己的端點。叢集也具有稱為「組態端點」的端點。如果您啟用自動探索並連線至組態端點，您的應用程式會自動「知道」每個節點端點，即使是在叢集新增或移除節點後。如需詳細資訊，請參閱 [自動探索](#)。

如需詳細資訊，請參閱 [端點](#)。

## ElastiCache 參數群組

快取參數群組是用於管理受支援引擎軟體之執行時間設定的便利方式。參數用於控制記憶體用量、移出政策、項目大小等等。ElastiCache 參數群組是可套用至叢集之引擎特定參數的具名集合。藉由執行此操作，您可以確保該叢集中的所有節點都以完全相同的方式設定。

如需支援的參數、其預設值及可修改的參數清單，請參閱 [DescribeEngineDefaultParameters \(describe-engine-default-parameters\)](#)。

如需 ElastiCache 參數群組的詳細資訊，請參閱 [使用參數群組設定引擎參數](#)。

## ElastiCache 安全

為了增強安全性，ElastiCache 節點存取僅限於在列入白名單的 Amazon EC2 執行個體上執行的應用程式。您可以使用安全群組來控制可存取叢集的 Amazon EC2 執行個體。

依預設，所有新 ElastiCache 叢集都會在 Amazon Virtual Private Cloud (Amazon VPC) 環境中啟動。您可以使用子網路群組授予叢集從特定子網路上所執行 Amazon EC2 執行個體進行存取的權限。如果您選擇在 Amazon VPC 之外執行叢集，您可以建立安全群組以授權在特定 Amazon EC2 安全群組中執行的 Amazon EC2 執行個體。

## ElastiCache 子網路群組

子網路群組是子網路的集合 (一般是私有)，您可以為在 Amazon Virtual Private Cloud (Amazon VPC) 環境中執行的叢集指定這些子網路。

如果您在 Amazon VPC 中建立叢集，則必須指定快取子網路群組。ElastiCache 使用該快取子網路群組來選擇該子網路內的子網路和 IP 位址，以便與快取節點建立關聯。

如需 Amazon VPC 環境中快取子網路群組使用方式的詳細資訊，請參閱 [Amazon VPC 和 ElastiCache 安全性、授權存取和子網路和子網路群組](#)。

## ElastiCache 對於內存緩存事件

當快取叢集上發生重大事件時，會 ElastiCache 將通知傳送至特定 Amazon SNS 主題。重大事件可能包含諸如新增節點失敗、新增節點成功、安全群組修改和其他事件。藉由監控重要的事件，您可以了解叢集目前的狀態，並根據事件採取正確的動作。

如需事件的詳細資 ElastiCache 訊，請參閱 [Amazon SNS 的 ElastiCache 事件監控](#)。

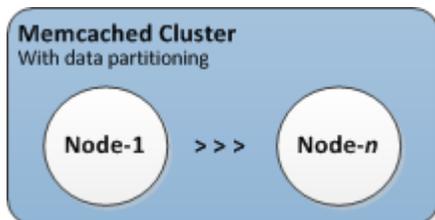
## 管理叢集

「叢集」是一或多個快取節點的集合，而這所有快取節點都執行 Memcached 快取引擎軟體的執行個體。在您建立叢集時，可以指定要使用的所有節點之引擎和版本。

下圖說明典型的 Memcached 叢集。Memcached 叢集包含 1 到 60 個節點，您可以在其中水平分割資料。

若要請求提高配額，請參閱 [AWS 服務配額](#)，並選擇配額類型 Nodes per cluster per instance type (每執行個體類型每叢集的節點數)。

典型的 Memcached 叢集如下所示。



大多數 ElastiCache 操作都是在集群級別執行的。您可以將叢集設定為含特定數量的節點和一個參數群組，以控制每個節點的屬性。叢集內的所有節點都設計為相同節點類型，並具備相同的參數和安全群組設定。

每個叢集都必須有一個叢集識別符。叢集識別符是客戶針對叢集提供的名稱。與 ElastiCache API 和 AWS CLI 命令進行交互時，此標識符指定特定的集群。該客戶在某個 AWS 區域中的叢集識別碼必須是唯一的。

ElastiCache 支持多種引擎版本。除非您有特定原因，否則我們建議您使用最新版本。

ElastiCache 叢集的設計是使用 Amazon EC2 執行個體存取。如果您在以 Amazon VPC 服務為基礎的 Virtual Private Cloud (VPC) 中啟動叢集，即可從 AWS 外部存取該叢集。如需詳細資訊，請參閱 [在 Amazon VPC 中存取 ElastiCache 快取的存取模式](#)。

如需受支援的 Memcached 版本清單，請參閱 [支援 ElastiCache 的 Memcached 版本](#)。

## 選擇網路類型

ElastiCache 支援網際網路通訊協定第 4 版和第 6 版 (IPv4 和 IPv6)，可讓您將叢集設定為接受：

- 只有 IPv4 連接，
- 只有 IPv6 連線，
- IPv4 和 IPv6 連線皆可 (雙堆疊)

在 [Nitro 系統](#) 上建置的所有執行個體中，使用 Memcached 引擎 1.6.6 版以上的工作負載都支援 IPv6。透過 IPv6 存取 ElastiCache 不另行收費。

### Note

不支援在 IPV6/雙堆疊可用性之前建立的叢集遷移。也不支援在新建立的叢集上切換網路類型。

## 設定網路類型的子網路

如果您在 Amazon VPC 中建立叢集，就必須指定子網路群組。ElastiCache 會使用此子網路群組選擇子網路及該子網路中的 IP 地址，以與您的節點建立關聯。ElastiCache 叢集需要具有 IPv4 和 IPv6 地址的雙堆疊子網路，以便在雙堆疊模式下運作，且僅限 IPv6 的子網路僅會以 IPv6 運作。

## 使用雙堆疊

當您建立快取叢集並選擇雙堆疊做為網路類型時，您需要指定 IP 探索類型 — IPv4 或 IPv6。ElastiCache 會將網路類型和 IP 探索預設為 IPv6，但這可以變更。如果您使用自動探索，只有您所選 IP 類型的 IP 地址會傳回至 Memcached 用戶端。

為了維持與所有現有用戶端的回溯相容性，我們會引入 IP 探索，讓您選取要在探索通訊協定中公告的 IP 類型 (即 IPv4 或 IPv6)。雖然這將自動探索限制為只有一種 IP 類型，但有了自動探索功能，雙堆疊仍然十分有用，因為可讓您從 IPv4 遷移 (或回復) 到 IPv6 探索 IP 類型，而不會導致停機。

## 啟用 TLS 的雙堆疊 ElastiCache 叢集

當針對 ElastiCache 叢集啟用 TLS 時，叢集探索功能 `config get cluster` 會傳回主機名稱而非 IP。接著會使用主機名稱而非 IP 來連線 ElastiCache 叢集，並執行 TLS 交握。這表示用戶端不會受到 IP 探索參數的影響。針對啟用 TLS 的叢集，IP 探索參數對偏好的 IP 通訊協定沒有影響。反之，用戶端在解析 DNS 主機名稱時偏好使用的 IP 通訊協定會決定所使用的 IP 通訊協定。

如需如何在解析 DNS 主機名稱時設定 IP 通訊協定偏好設定的範例，請參閱 [啟用 TLS 的雙堆疊 ElastiCache 叢集](#)。

## 使用 AWS Management Console

使用 AWS Management Console 建立快取叢集時，在 Connectivity (連線) 下，選擇網路類型，可以是 IPv4、IPv6 或 Dual stack (雙堆疊)。若您選擇雙堆疊，則須選取 IPv6 或 IPv4 做為 Discovery IP type (探索 IP 類型)。

如需更多詳細資訊，請參閱 [建立 Memcached 叢集 \(主控台\)](#)。

## 使用 CLI

使用 CLI 建立快取叢集時，您可以使用 [create-cache-cluster](#) 命令並指定 NetworkType 和 IPDiscovery 參數：

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

## 資料分層

組成複寫群組且使用 r6gd 系列節點類型的叢集會在記憶體和本機 SSD (固態硬碟) 儲存之間進行資料分層。除了將資料存放在記憶體中之外，資料分層還能藉由在每個叢集節點中使用成本較低的固態硬碟

(SSD)，為 Redis 工作負載提供全新的價格效能方案。非常適合定期存取高達 20% 的整體資料集的工作負載，以及在存取 SSD 資料時可容忍額外延遲的應用程式。

在具有資料分層的叢集上，ElastiCache 監視其儲存之每個項目的上次存取時間。當可用記憶體 (DRAM) 完全消耗時，會 ElastiCache 使用最近最少使用的 (LRU) 演算法，將不常存取的項目從記憶體自動移至 SSD。隨後存取 SSD 上的資料時，在處理要求之前，ElastiCache 會自動且非同步地將其移回記憶體。如果您的工作負載只會定期存取其資料的子集，則資料分層是以符合成本效益的方式擴展容量的最佳方式。

請注意，使用資料分層時，金鑰本身一律會保留在記憶體中，而 LRU 則會控制記憶體與磁碟上值的位置。一般而言，建議在使用資料分層時，金鑰大小小於您值的大小。

資料分層專為盡量降低對應用程式工作負載的效能影響所設計。例如，假設 500 個位元組的字串值，與請求在記憶體中的資料相比，可以預期請求存放在 SSD 上的資料時會有額外 300 微秒的延遲。

使用最大的資料分層節點大小 (cache.r6gd.16xlarge)，可以在單一 500 個節點的叢集中存放高達 1 PB (使用 1 個僅供讀取複本時為 500 TB)。資料分層與中支援的所有 Redis 命令和資料結構相容。ElastiCache 不需要任何用戶端變更就能使用此功能。

## 自動識別叢集中的節點

針對執行 Memcached 引擎的叢集，ElastiCache 支援自動探索功能，此功能可讓用戶端程式自動識別快取叢集中的所有節點，並初始化及維持與這些節點的連線。

### Note

自動探索是專為在 Amazon ElastiCache Memcached 上執行的快取叢集所新增。

透過自動探索，您的應用程式便不需要手動連線到個別快取節點；您的應用程式可改為連線到單一 Memcached 節點，並擷取節點清單。藉由該清單，您的應用程式便會感知到叢集中的其餘節點，並可以連線到其中任何一個節點。您不需要在應用程式中硬式編碼個別快取節點端點。

若您在叢集上使用雙堆疊網路類型，則視您選取的地址，自動探索只會傳回 IPv4 或 IPv6 地址。如需詳細資訊，請參閱 [選擇網路類型](#)。

叢集中所有快取節點都會維持一份其他所有節點的中繼資料清單。此中繼資料會在將節點新增到叢集或從叢集中移除節點時更新。

## 主題

- [自動探索的優點](#)
- [自動探索的運作方式](#)
- [使用自動探索](#)
- [手動連線至快取節點](#)
- [將自動探索新增到您的用戶端程式庫](#)
- [具有自動探索功能的 ElastiCache 用戶端](#)

## 自動探索的優點

自動探索可提供下列優點：

- 當您增加快取叢集中的節點數時，新的節點會自行向組態端點及所有其他節點註冊。當您從快取叢集移除節點時，離開的節點會自行取消註冊。在這兩種情況下，叢集中所有其他的節點都會使用最新的快取節點中繼資料更新。
- 自動偵測快取節點故障，並會自動取代失敗的節點。

### Note

在完成更換節點之前，節點會繼續失敗。

- 用戶端程式只需要連線到組態端點。之後，自動探索程式庫便會連線到叢集內所有其他的節點。
- 用戶端程式會每分鐘輪詢叢集一次 (此間隔可視需要進行調整)。若叢集組態發生任何變更 (例如新增節點或刪除節點)，用戶端便會接收到更新後的中繼資料清單。然後，用戶端便會視需要連線 (或中斷連線) 到這些節點。

自動探索功能會在所有 ElastiCache Memcached 快取叢集上啟用。您不需要重新啟動任何快取節點，便可使用這項功能。

## 自動探索的運作方式

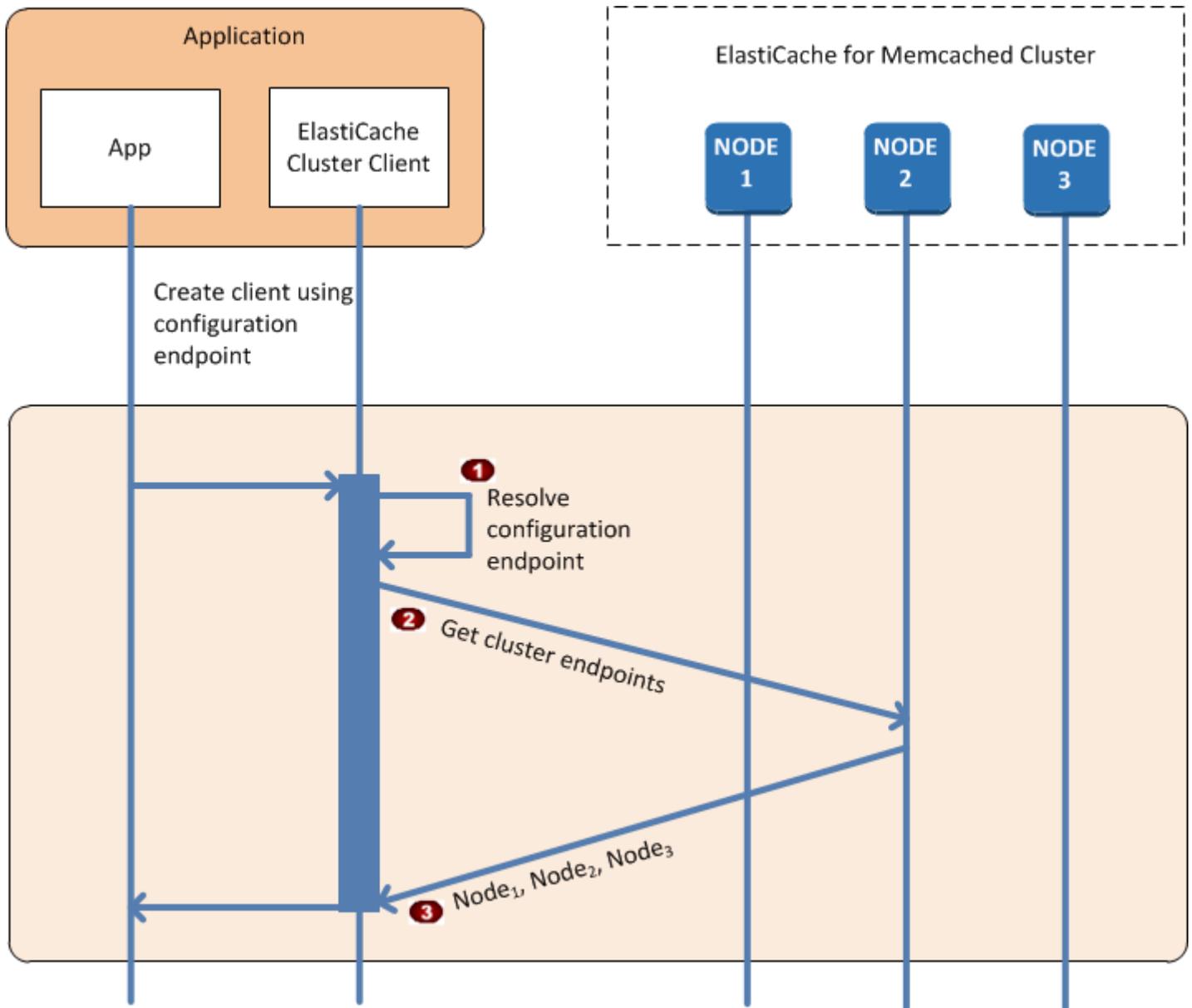
### 主題

- [連線至快取節點](#)
- [一般叢集操作](#)
- [其他操作](#)

本節說明用戶端應用程式使用 ElastiCache 叢集用戶端管理快取節點連線，以及與快取中資料項目互動的方式。

### 連線至快取節點

從應用程式的觀點而言，連線到叢集組態端點與直接連線到個別快取節點沒什麼不同。以下順序圖表顯示連線到快取節點的程序。



### 連線至快取節點的程序

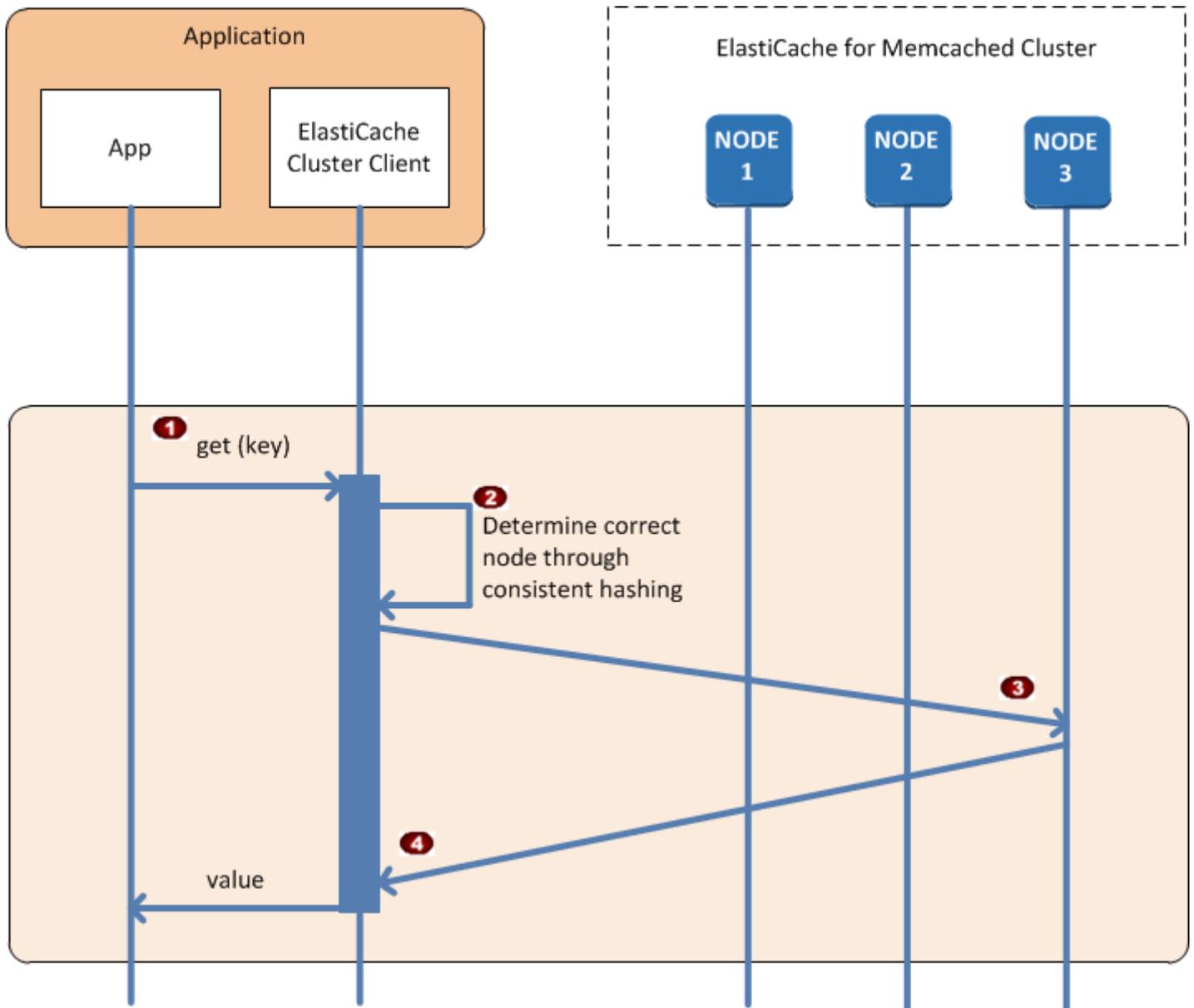
- 應用程式解析組態端點的 DNS 名稱。因為組態端點包含所有快取節點的 CNAME 項目，DNS 名稱會解析到其中一個節點。用戶端接著便能連線到該節點。
- 用戶端請求所有其他節點的組態資訊。因為每個節點都會維持叢集中所有節點的組態資訊，任何節點都能在請求時將組態資訊傳遞給用戶端。
- 用戶端接收到目前快取節點主機名稱及 IP 地址的清單。然後，它便能連線到叢集中所有其他的節點。

**Note**

用戶端程式每分鐘更新快取節點主機名稱及 IP 地址的清單一次。此輪詢間隔可視需要進行調整。

**一般叢集操作**

應用程式連線到所有快取節點之後，ElastiCache 叢集用戶端便會判斷哪些節點應存放個別資料項目，以及稍後應對哪些節點查詢該資料項目。以下順序圖表顯示一般叢集操作的程序。



## 一般叢集操作的程序

- 應用程式針對個別資料項目發出 get 請求，以資料項目的鍵進行識別。
- 用戶端針對鍵使用雜湊演算法，判斷哪個快取節點包含該資料項目。
- 隨即從適當節點請求資料項目。
- 資料項目傳回應用程式。

## 其他操作

在某些情況下，您可能會對叢集的節點進行變更。例如，您可以新增額外節點以滿足其他需求，或刪除節點以在需求減少期間節省成本。或者，您可能會因為任何節點故障而替換掉節點。

當叢集內發生變更，使中繼資料需要更新叢集的端點時，便會同時變更所有節點。因此，任何指定節點中的中繼資料都會與叢集內其他所有節點中的中繼資料保持一致。

在其中任何一種情況下，由於叢集內所有節點的中繼資料都會同時更新，因此所有節點內的中繼資料都會隨時保持一致。建議您一律使用組態端點來取得叢集內各種節點的端點。透過使用組態端點，您可以確保不會從在您面前「消失」的節點取得端點資料。

## 新增節點

節點啟動期間，中繼資料內不會包含其端點。當節點可供使用時，它便會新增到每個叢集節點的中繼資料。在此案例中，所有節點內的中繼資料都會保持一致，並且您只能在新節點可供使用之後與它互動。在節點可供使用之前，您將無法得知它，並且您會以相同方式與叢集內的節點互動，就好像新節點不存在。

## 刪除節點

移除節點時，會先從中繼資料移除其端點，然後再從叢集移除節點。在此案例中，所有節點內的中繼資料都會保持一致，並且不會在節點無法使用的同時包含要移除節點的端點。在移除節點期間，中繼資料內不會報告該節點，因此您的應用程式屆時只會與  $n-1$  個剩餘節點進行互動，宛如節點不存在。

## 取代節點

若節點失敗，ElastiCache 會撤下該節點，並啟用替代節點。取代程序可能需要幾分鐘。在此期間，所有節點內的中繼資料仍會顯示失敗節點的端點，但任何嘗試與該節點進行的互動都會失敗。因此，建議您的邏輯一律包含重試邏輯。

## 使用自動探索

若要開始使用自動探索，請遵循以下步驟：

- [步驟 1：取得端點組態](#)
- [步驟 2：下載 ElastiCache 叢集用戶端](#)
- [步驟 3：修改您的應用程式](#)

### 步驟 1：取得端點組態

為了連線到叢集，用戶端程式必須知道叢集的組態端點。請參閱主題：[尋找叢集的端點 \(主控台\)](#)

您也可以使用 `aws elasticache describe-cache-clusters` 命令搭配 `--show-cache-node-info` 參數：

無論您用來尋找叢集端點的方法為何，組態端點的地址內都會一律包含 `.cfg`。

Example 使用 ElastiCache 的 AWS CLI 來尋找端點

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

此作業會產生類似下列的輸出 (JSON 格式)：

```
{  
  "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"
}

```

```
]
}
```

## 步驟 2：下載 ElastiCache 叢集用戶端

若要利用自動探索功能，用戶端程式必須使用 ElastiCache 叢集用戶端。ElastiCache 叢集用戶端適用於 Java、PHP 及 .NET，並包含探索及連線到所有快取節點的必要邏輯。

### 下載 ElastiCache 叢集用戶端

1. 登入 AWS 管理主控台，並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從 ElastiCache 主控台中，選擇 ElastiCache Cluster Client (ElastiCache 叢集用戶端)，然後選擇 Download (下載)。

如需適用於 Java 的 ElastiCache 叢集用戶端原始碼，請前往 <https://github.com/amazonwebservices/aws-elasticache-cluster-client-memcached-for-java>。此程式庫是以熱門的 Spymemcached 用戶端為基礎。ElastiCache 叢集用戶端是根據 Amazon 軟體授權 (<https://aws.amazon.com/asl>) 發行。您可以視需要自由修改來源碼。您甚至可以將程式碼併入其他開放原始碼 Memcached 程式庫，或是併入您自己的用戶端程式碼。

#### Note

若要使用適用於 PHP 的 ElastiCache 叢集用戶端，您必須先在 Amazon EC2 執行個體上安裝它。如需更多詳細資訊，請參閱 [安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)。  
若為支援 TLS 的用戶端，請下載 PHP 7.4 或更高版本的二進位檔。  
若要使用適用於 .NET 的 ElastiCache 叢集用戶端，您必須先在 Amazon EC2 執行個體上安裝它。如需更多詳細資訊，請參閱 [安裝適用於 .NET 的 ElastiCache 叢集用戶端](#)。

## 步驟 3：修改您的應用程式

修改您的應用程式以使用自動探索。以下各節示範如何使用適用於 Java、PHP 及 .NET 的 ElastiCache 叢集用戶端。

#### Important

指定叢集的組態端點時，請務必確認端點的地址內包含 ".cfg"，如以下所示。請不要使用未包含 ".cfg" 的 CNAME 或端點。

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

明確指定叢集組態端點失敗，可能會導致設定至特定節點。

## 使用適用於 Java 的 ElastiCache 叢集用戶端

以下程式會示範如何使用 ElastiCache 叢集用戶端連線到叢集組態端點，然後將資料項目新增到快取。使用自動探索，程式便會連線到叢集內的所有節點，而不會進行其他介入。

```
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");
    }
}
```

## 使用適用於 PHP 的 ElastiCache 叢集用戶端

以下程式會示範如何使用 ElastiCache 叢集用戶端連線到叢集組態端點，然後將資料項目新增到快取。使用自動探索，程式將會連線到叢集內的所有節點，而不會進行其他介入。

若要使用適用於 PHP 的 ElastiCache 叢集用戶端，您必須先在 Amazon EC2 執行個體上安裝它。如需詳細資訊，請參閱 [安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)

```
<?php

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

/* Configuration endpoint to use to initialize memcached client.
 * This is only an example. */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the ElastiCache cluster.
 * This is only an example */
$server_port = 11211;

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

/**
 * Store the data for 60 seconds in the cluster.
 * The client will decide which cache host will store this item.
```

```
*/
$dynamic_client->set('key', 'value', 60);

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

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

如需如何使用已啟用 TLS 的 ElastiCache 叢集用戶端範例，請參閱[使用傳輸中加密搭配 PHP 和 Memcached](#)。

使用適用於 .NET 的 ElastiCache 叢集用戶端

#### Note

自 2022 年 5 月起，已淘汰 ElastiCache .NET 叢集用戶端。

適用於 ElastiCache 的 .NET 用戶端為開放原始碼資源，位於 <https://github.com/aws-labs/elasticache-cluster-config-net>。

.NET 應用程式通常會從其組態檔取得組態。以下是範例應用程式組態檔。

```
<?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>
```

以下 C# 程式示範如何使用 ElastiCache 叢集用戶端連線到叢集組態端點，然後將資料項目新增到快取。使用自動探索，程式將會連線到叢集內的所有節點，而不會進行其他介入。

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

## 手動連線至快取節點

若您的用戶端程式未使用自動探索，它可以手動連線到每個快取節點。這是 Memcached 用戶端的預設行為。

您可以從 [AWS 管理主控台](#) 取得快取節點主機名稱及連接埠號碼的清單。您也可以使用 AWS CLI `aws elasticache describe-cache-clusters` 命令，搭配 `--show-cache-node-info` 參數。

### Example

以下 Java 程式碼片段示範如何連線到四節點快取叢集內的所有節點：

```
...

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

若您透過新增或移除節點向上或向下擴展快取叢集，您將需要在用戶端程式碼中更新節點清單。

## 將自動探索新增到您的用戶端程式庫

自動探索的組態資訊會備援存放在每個快取叢集節點。用戶端應用程式可查詢任何快取節點，並取得叢集內所有節點的組態資訊。

應用程式執行此作業的方式取決於快取引擎版本：

- 若快取引擎版本為 1.4.14 或更高版本，請使用 `config` 命令。
- 若快取引擎版本低於 1.4.14，請使用 `get AmazonElastiCache:cluster` 命令。

這兩個命令的輸出完全相同，並且會在以下的[輸出格式](#)一節中說明。

### 1.4.14 或更高的快取引擎版本

針對 1.4.14 或更高的快取引擎版本，請使用 `config` 命令。此命令已新增到 ElastiCache 的 Memcached ASCII 及二進位通訊協定中，並且已在 ElastiCache 叢集用戶端中實作。若您希望搭配其他用戶端程式庫使用自動探索，該程式庫必須進行擴充以支援 `config` 命令。

#### Note

以下文件適用於 ASCII 通訊協定。但是，`config` 命令同時支援 ASCII 及二進位。若您希望新增支援使用二進位通訊協定的自動探索功能，請參閱 [ElastiCache 叢集用戶端的原始碼](#)。

## Syntax (語法)

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

### 選項

名稱	描述	必要
sub-command	用來與快取節點互動的子命令。針對自動探索，此子命令為 <code>get</code> 。	是
key	鍵名稱，叢集組態會存放在該鍵之下。針對自動探索，此鍵名為 <code>cluster</code> 。	是

若要取得叢集組態資訊，請使用以下命令：

```
config get cluster
```

#### 1.4.14 以下的快取引擎版本

若要取得叢集組態資訊，請使用以下命令：

```
get AmazonElastiCache:cluster
```

#### Note

請勿竄改 "AmazonElastiCache:cluster" 鍵，因為這是保存叢集組態資訊的位置。若您覆寫了此鍵，則不正確的用戶端設定可能持續短暫的期間 (不超過 15 秒)，之後 ElastiCache 便會自動並正確地更新組態資訊。

#### 輸出格式

無論您使用 `config get cluster` 或 `get AmazonElastiCache:cluster`，回應都由兩行文字組成：

- 組態資訊的版本編號。每當新增節點至快取叢集，或從快取叢集移除節點時，版本編號都會加一。
- 快取節點的清單。清單中的每個節點都會以 `hostname|ip-address|port` 群組表示，並且每個節點都會以一個空格分隔。

歸位字元和換行字元 (CR + LF) 會出現在每一行的結尾。資料行的結尾包含一個換行字元 (LF)，CR + LF 會新增到該處。組態版本行會以不帶 CR 的 LF 終止。

包含三個節點的快取叢集會以以下方式表示：

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

每個節點都會同時顯示 CNAME 及私有 IP 地址。CNAME 一律會存在。若私有 IP 地址無法使用，便不會顯示。但是，仍會印出管道字元 "|".

#### Example

以下是當您查詢組態資訊時，所傳回承載的範例：

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

### Note

- 第二行指出組態資訊至今已經過修改十二次。
- 在第三行中，節點清單是以主機名稱的字母順序排序。此順序的排列方式可能會跟您目前於用戶端應用程式中使用的順序不同。

## 具有自動探索功能的 ElastiCache 用戶端

本節說明如何安裝及設定 ElastiCache PHP 和 .NET 用戶端。

### 主題

- [安裝及編譯叢集用戶端](#)
- [設定 ElastiCache 用戶端](#)

### 安裝及編譯叢集用戶端

本節介紹安裝、設定及編譯 PHP 和 .NET Amazon ElastiCache 自動探索叢集用戶端的內容。

### 主題

- [安裝適用於 .NET 的 ElastiCache 叢集用戶端](#)
- [安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)
- [編譯適用於 PHP 的 ElastiCache 叢集用戶端原始碼](#)

### 安裝適用於 .NET 的 ElastiCache 叢集用戶端

### Note

自 2022 年 5 月起，已淘汰 ElastiCache .NET 叢集用戶端。

適用於 ElastiCache .NET 叢集用戶端的程式碼為開放原始碼，如有需要請前往 <https://github.com/awslabs/elasticache-cluster-config-net>。

本節說明如何在 Amazon EC2 執行個體上安裝、更新及移除適用於 ElastiCache 叢集用戶端的 .NET 元件。如需自動探索的詳細資訊，請參閱 [自動探索](#)。如需使用用戶端的 .NET 程式碼範例，請參閱 [使用 DotNet 的自動探索](#)。

## 主題

- [安裝 .NET](#)
- [下載適用於 ElastiCache 的 ElastiCache .NET 叢集用戶端](#)
- [使用 NuGet 安裝 AWS 組件](#)

## 安裝 .NET

您必須安裝 .NET 3.5 或更新版本，才能使用適用於 ElastiCache 的 AWS .NET 軟體開發套件。如果您未安裝 .NET 3.5 或更新版本，您可以從 <http://www.microsoft.com/net> 下載及安裝最新版本。

## 下載適用於 ElastiCache 的 ElastiCache .NET 叢集用戶端

### 下載 ElastiCache .NET 叢集用戶端

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格上，按一下 ElastiCache Cluster Client (ElastiCache 叢集用戶端)。
3. 在 Download ElastiCache Memcached Cluster Client (下載 ElastiCache Memcached 叢集用戶端) 清單中，選取 .NET，然後按一下 Download (下載)。

## 使用 NuGet 安裝 AWS 組件

NuGet 是適用於 .NET 平台的套件管理系統。NuGet 知道組件的依存性，並會自動安裝所需的檔案。NuGet 安裝的組件會與您的解決方案一同存放，而非位於一個中央位置 (例如 Program Files)。這可讓您安裝應用程式的特定組件版本，而不會產生相容性問題。

## 安裝 NuGet

NuGet 可從 MSDN 上的安裝資源庫安裝。請參閱 <https://visualstudiogallery.msdn.microsoft.com/27077b70-9dad-4c64-adcf-c7cf6bc9970c>。若您使用 Visual Studio 2010 或更新版本，NuGet 便已自動安裝。

您可以從 Solution Explorer (方案總管) 或 Package Manager Console (套件管理員主控台) 使用 NuGet。

從方案總管使用 NuGet

在 Visual Studio 2010 中從方案總管使用 NuGet

1. 從 Tools (工具) 選單，選取 Library Package Manager (程式庫套件管理員)。
2. 按一下 Package Manager Console (套件管理員主控台)。

在 Visual Studio 2012 或 Visual Studio 2013 中從方案總管使用 NuGet

1. 從 Tools (工具) 選單，選取 NuGet Package Manager (NuGet 套件管理員)。
2. 按一下 Package Manager Console (套件管理員主控台)。

從命令列，您可以使用 Install-Package 安裝組件，如下所示。

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

若要查看列出可透過 NuGet 取得的所有套件之頁面 (例如 AWSSDK 和 AWS.Extensions 組件)，請參閱 NuGet 網站 (<http://www.nuget.org>)。每個套件的頁面都包含一個使用主控台安裝套件的範例命令列，以及一份可透過 NuGet 取得的先前版本套件清單。

如需 Package Manager Console (套件管理員主控台) 命令的詳細資訊，請參閱 <http://nuget.codeplex.com/wikipage?title=Package%20Manager%20Console%20Command%20Reference%20%28v1.3%29>。

## 安裝適用於 PHP 的 ElastiCache 叢集用戶端

本節說明如何在 Amazon EC2 執行個體上安裝、更新及移除適用於 ElastiCache 叢集用戶端的 PHP 元件。如需自動探索的詳細資訊，請參閱[自動識別叢集中的節點](#)。如需要使用用戶端的範例 PHP 程式碼，請參閱[使用適用於 PHP 的 ElastiCache 叢集用戶端](#)。

### 主題

- [下載安裝套件](#)
- [針對已安裝 php-memcached 延伸的使用者](#)
- [新使用者的安裝步驟](#)
- [移除 PHP 叢集用戶端](#)

### 下載安裝套件

為了確保所使用之適用於 PHP 的 ElastiCache 叢集用戶端是正確版本，您必須知道 Amazon EC2 執行個體上安裝的 PHP 版本。您還必須知道 Amazon EC2 執行個體是在 64 位元還是 32 位元版本的 Linux 上執行。

### 判斷 Amazon EC2 執行個體上安裝的 PHP 版本

- 在命令提示中，執行下列命令：

```
php -v
```

PHP 版本將顯示在輸出中，如下列範例所示：

```
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

如果 PHP 與 Memcached 版本不相容，您將收到類似下列內容的錯誤訊息：

```
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
```

如果發生這種情況，您需要從來源碼編譯模組。如需更多詳細資訊，請參閱 [編譯適用於 PHP 的 ElastiCache 叢集用戶端原始碼](#)。

### 判斷 Amazon EC2 AMI 架構 (64 位元或 32 位元)

1. 請登入 AWS Management Console，並在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
2. 在 Instances (執行個體) 清單中，按一下 Amazon EC2 執行個體。
3. 在 Description (描述) 標籤中，查看 AMI: (AMI : ) 欄位。64 位元執行個體應該使用 x86\_64 做為描述的一部分；若為 32 位元執行個體，請在此欄位中尋找 i386 或 i686。

您現在可以下載 ElastiCache 叢集用戶端。

### 下載適用於 PHP 的 ElastiCache 叢集用戶端

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從 ElastiCache 主控台選擇 ElastiCache 叢集用戶端。
3. 從 Download ElastiCache Memcached Cluster Client (下載 ElastiCache Memcached 叢集用戶端) 清單中，選擇符合您 PHP 版本和 AMI 架構的 ElastiCache 叢集用戶端，然後選擇 Download (下載) 按鈕。

若為支援 TLS 的用戶端，請下載 PHP 7.4 或更高版本的二進位檔。

### 針對已安裝 php-memcached 延伸的使用者

#### 更新 php-memcached 安裝

1. 移除先前安裝之適用於 PHP 的 Memcached 延伸，如 [移除 PHP 叢集用戶端](#) 主題所述。
2. 安裝新的 ElastiCache php-memcached 延伸，如先前「[新使用者的安裝步驟](#)」中所述。

### 新使用者的安裝步驟

#### 主題

- [為新使用者安裝 PHP 7.x - 8.x](#)
- [為新使用者安裝 PHP 5.x](#)

為新使用者安裝 PHP 7.x - 8.x

### 主題

- [在 Amazon Linux 2 AMI 上安裝 PHP 7.x - 8.x](#)
- [在 Amazon Linux 201609 AMI 上安裝 PHP 7.x - 8.x](#)
- [在 SUSE Linux 15 AMI 上安裝 PHP 7.x - 8.x](#)
- [在 Ubuntu 22.04 AMI 上安裝 PHP 7.x - 8.x](#)

在 Amazon Linux 2 AMI 上安裝 PHP 7.x - 8.x

#### Note

必要時，以您目前使用的版本取代 *PHP-7.x*。

1. 從 AMI 啟動新的執行個體。
2. 執行以下命令：

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

3. 使用 `amazon-linux-extras` 安裝 PHP 7.x

使用 Amazon Linux 2 時，您可用 Extras Library 將應用程式和軟體更新安裝至執行個體。這些軟體更新即為主題。您可安裝特定版本的主題，或略過版本資訊，以使用最新版本。如需詳細資訊，請參閱 [Extras Library \(Amazon Linux 2\)](#)。

如要執行此作業，請依照下列步驟進行；

- a. 首先，確認 `amazon-linux-extras` 是否已安裝。
- b. 如果未安裝，請使用下列命令來安裝：

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

- c. 確認 PHP 7.x 主題在 Amazon Linux 2 機器上可用：

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

- d. 在輸出中查看所有 PHP 7 主題，並選擇您想要的版本：

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

- e. 從儲存庫安裝 PHP 套件。例如：

```
sudo yum clean metadata
```

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

4. 下載 Amazon ElastiCache 叢集用戶端。

- 前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。

在 ElastiCache 儀表板下，前往 ElastiCache Cluster Client (ElastiCache 叢集用戶端)，然後選擇您想要的 PHP7 版本。

- 從命令列，以所需 PHP 版本取代 PHP-7.X，並以所需架構 (X86 或 ARM) 取代 ARCH，若 PHP  $\geq 7.4$ ，以所需 OpenSSL 版本 (openssl1.1 或 openssl3) 取代 OpenSSL。如果您使用 PHP  $> 7.4$ ，請移除 OpenSSL 後綴。

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

5. 使用 `tar -zxvf` 解壓縮下載的檔案。

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

6. 使用 root 許可，將解壓縮的成品檔案 `amazon-elasticache-cluster-client.so` 複製到 `/usr/lib64/php/modules`。

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

7. 將 `extension=amazon-elasticache-cluster-client.so` 新增到 `/etc/php.ini` 檔案中
8. 如果您已下載 PHP 7.4 或更高版本的 ElastiCache 叢集用戶端，請安裝 OpenSSL 1.1.x 或更高版本。OpenSSL 1.1.1 的安裝說明：

```
sudo yum -y update  
sudo yum install -y make gcc perl-core pcre-devel wget zlib-devel  
wget https://www.openssl.org/source/openssl-1.1.1c.tar.gz  
tar xvf openssl-1.1.1c.tar.gz
```

```
cd openssl-1.1.1c
./config
make
sudo make install
sudo ln -s /usr/local/lib64/libssl.so.1.1 /usr/lib64/libssl.so.1.1
```

## 在 Amazon Linux 201609 AMI 上安裝 PHP 7.x - 8.x

### Note

必要時，以您目前使用的版本取代 *php7.x*。

1. 從 AMI 啟動新的執行個體。如需如何執行此動作的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [步驟 1：啟動執行個體](#)。

2. 執行以下命令：

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

3. 安裝 PHP

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

4. 下載 Amazon ElastiCache 叢集用戶端。

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

5. 將 latest-64bit 解壓縮。

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

6. 使用 root 許可，將解壓縮的成品檔案 `amazon-elasticache-cluster-client.so` 複製到 `/usr/lib64/php/7.x/modules/`。

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

7. 建立 `50-memcached.ini` 檔案。

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

8. 啟動或重新啟動您的 Apache 伺服器。

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

## 在 SUSE Linux 15 AMI 上安裝 PHP 7.x - 8.x

### Note

必要時，以您目前使用的版本取代 *php7.x*。

1. 從 AMI 啟動新的執行個體。
2. 執行以下命令：

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

3. 安裝 PHP

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

或

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

4. 下載 Amazon ElastiCache 叢集用戶端，以所需架構 (X86 或 ARM) 取代 <ARCH>。SUSE 15 內建 OpenSSL1.1，因此若 PHP >= 7.4，請選擇具 OpenSSL1 的用戶端二進位檔。如果您使用 PHP < 7.4，請移除 OpenSSL 後綴。

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

## 5. 將 latest-64bit 解壓縮。

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

## 6. 使用 root 許可，將解壓縮的成品檔案 amazon-elasticache-cluster-client.so 複製到 /usr/lib64/php7/extensions/。

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

## 7. 將 extension=amazon-elasticache-cluster-client.so 這一行插入到 /etc/php7/cli/php.ini 檔案。

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

## 8. 啟動或重新啟動您的 Apache 伺服器。

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

## 在 Ubuntu 22.04 AMI 上安裝 PHP 7.x - 8.x

### Note

必要時，以您目前使用的版本取代 *php7.x*。

1. 從 AMI 啟動新的執行個體。
2. 執行以下命令：

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

## 3. 安裝 PHP

- a. PHP 8.1 的安裝說明：

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

## b. PHP 7.4 的安裝說明：

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

4. 下載 Amazon ElastiCache 叢集用戶端，以所需架構 (X86 或 ARM) 取代 <ARCH>。Ubuntu 22.04 內建 OpenSSL3，因此若 PHP >= 7.4，請選擇具 OpenSSL3 的用戶端二進位檔。如果您使用 PHP < 7.4，請移除 OpenSSL 後綴。

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

5. 將 latest-64bit 解壓縮。

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

6. 使用 root 許可，將解壓縮的成品檔案 amazon-elasticache-cluster-client.so 複製到 php 擴充目錄 /usr/lib/php/20190902。如果沒有該擴充目錄，您可以執行以下命令找到：php -i | grep extension\_dir
7. 將 extension=amazon-elasticache-cluster-client.so 這一行插入到 /etc/php/7.x/cli/php.ini 檔案。

## 為新使用者安裝 PHP 5.x

### 主題

- [在 Amazon Linux AMI 2014.03 \(64 位元及 32 位元\) 上安裝 PHP 5](#)
- [在 Red Hat Enterprise Linux 7.0 AMI \(64 位元及 32 位元\) 上安裝 PHP 5](#)
- [在 Ubuntu Server 14.04 LTS AMI \(64 位元及 32 位元\) 上安裝 PHP 5](#)
- [為 SUSE Linux Enterprise Server 11 AMI \(64 位元或 32 位元\) 安裝 PHP 5](#)
- [其他 Linux 發行版本](#)

## 在 Amazon Linux AMI 2014.03 (64 位元及 32 位元) 上安裝 PHP 5

1. 啟動 Amazon Linux 執行個體 (64 位元或 32 位元)，並登入其中。

## 2. 安裝 PHP 相依性：

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

## 3. 下載適用於您 Amazon EC2 執行個體和 PHP 版本的正確 php-memcached 套件。如需更多詳細資訊，請參閱 [下載安裝套件](#)。

## 4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑：

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

以下是 PHP 5.4、64 位元 Linux 的範例安裝命令。在此範例中，將 *X.Y.Z* 取代為實際版本號碼：

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

### Note

務必使用安裝成品的最新版本。

## 5. 使用 root/sudo 許可，在 /etc/php.d 目錄中新增名為 memcached.ini 的新檔案，並在該檔案中插入 "extension=amazon-elasticache-cluster-client.so"：

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

## 6. 啟動或重新啟動您的 Apache 伺服器。

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

## 在 Red Hat Enterprise Linux 7.0 AMI (64 位元及 32 位元) 上安裝 PHP 5

1. 啟動 Red Hat Enterprise Linux 執行個體 (64 位元或 32 位元)，並登入其中。
2. 安裝 PHP 相依性：

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

3. 下載適用於您 Amazon EC2 執行個體和 PHP 版本的正確 php-memcached 套件。如需更多詳細資訊，請參閱 [下載安裝套件](#)。
4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑：

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

5. 使用 root/sudo 許可，在 /etc/php.d 目錄中新增名為 memcached.ini 的新檔案，並在該檔案中插入 extension=amazon-elasticache-cluster-client.so。

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

6. 啟動或重新啟動您的 Apache 伺服器。

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

在 Ubuntu Server 14.04 LTS AMI (64 位元及 32 位元) 上安裝 PHP 5

1. 啟動 Ubuntu Linux 執行個體 (64 位元或 32 位元)，並登入其中。
2. 安裝 PHP 相依性：

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

3. 下載適用於您 Amazon EC2 執行個體和 PHP 版本的正確 php-memcached 套件。如需更多詳細資訊，請參閱 [下載安裝套件](#)。
4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑。

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

#### Note

此安裝步驟會將建置成品 amazon-elasticache-cluster-client.so 安裝到 /usr/lib/php5/20121212\* 目錄。驗證組建成品的絕對路徑，因為您在下一個步驟中需要它。

如果上一個命令無法運作，您需要從下載的 \*.tgz 檔案以手動方式解壓縮 PHP 用戶端成品 amazon-elasticache-cluster-client.so，並將其複製到 /usr/lib/php5/20121212\* 目錄。

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

5. 使用 root/sudo 許可，在 /etc/php5/cli/conf.d 目錄中新增名為 memcached.ini 的新檔案，並在該檔案中插入 "extension=<amazon-elasticache-cluster-client.so 的絕對路徑>"。

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

6. 啟動或重新啟動您的 Apache 伺服器。

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

為 SUSE Linux Enterprise Server 11 AMI (64 位元或 32 位元) 安裝 PHP 5

1. 啟動 SUSE Linux 執行個體 (64 位元或 32 位元)，並登入其中。
2. 安裝 PHP 相依性：

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

3. 下載適用於您 Amazon EC2 執行個體和 PHP 版本的正確 php-memcached 套件。如需更多詳細資訊，請參閱 [下載安裝套件](#)。
4. 安裝 php-memcached。URI 應該是安裝套件的下載路徑。

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

5. 使用 root/sudo 許可，在 /etc/php5/conf.d 目錄中新增名為 memcached.ini 的新檔案，並在該檔案中插入 **extension=amazon-elasticache-cluster-client.so**。

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

## 6. 啟動或重新啟動您的 Apache 伺服器。

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

### Note

如果步驟 5 對先前的任何平台沒有作用，請驗證 `amazon-elasticache-cluster-client.so` 的安裝路徑。同時，請指定延伸中二進位檔的完整路徑。此外，請驗證使用中的 PHP 是支援的版本。我們支援 5.3 版到 5.5 版。

## 其他 Linux 發行版本

在某些系統上 (特別是 CentOS7 及 Red Hat Enterprise Linux (RHEL) 7.1)，`libsasl2.so.3` 已取代 `libsasl2.so.2`。在這些系統上，當您載入 ElastiCache 叢集用戶端時，它會嘗試尋找並載入 `libsasl2.so.2`，但這項嘗試失敗。若要解決此問題，請建立指向 `libsasl2.so.3` 的符號連結，以便在用戶端嘗試載入 `libsasl2.so.2` 時，將其重新導向至 `libsasl2.so.3`。下列程式碼會建立此符號連結。

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

## 移除 PHP 叢集用戶端

### 主題

- [移除舊版的 PHP 7 或更高版本](#)
- [移除舊版的 PHP 5](#)

## 移除舊版的 PHP 7 或更高版本

## 移除舊版的 PHP 7 或更高版本

1. 從適當的 PHP lib 目錄中移除 `amazon-elasticache-cluster-client.so` 檔案，如之前在安裝指示中所述。請參閱[針對已安裝 php-memcached 延伸的使用者](#)中您的安裝所適用章節。
2. 從 `php.ini` 檔案中移除 `extension=amazon-elasticache-cluster-client.so` 這一行。

### 3. 啟動或重新啟動您的 Apache 伺服器。

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

#### 移除舊版的 PHP 5

#### 移除舊版的 PHP 5

##### 1. 移除 php-memcached 延伸：

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

##### 2. 移除適當目錄中所新增的 memcached.ini 檔案，如先前安裝步驟中所述。

## 編譯適用於 PHP 的 ElastiCache 叢集用戶端原始碼

本節說明如何取得及編譯適用於 PHP 的 ElastiCache 叢集用戶端之原始碼。

您需要從 GitHub 提取兩個套件並進行編譯：[aws-elasticache-cluster-client-libmemcached](#) 及 [aws-elasticache-cluster-client-memcached-for-php](#)。

### 主題

- [編譯 libmemcached 程式庫](#)
- [編譯適用於 PHP 的 ElastiCache Memcached 自動探索用戶端](#)

### 編譯 libmemcached 程式庫

#### 必要程式庫

- OpenSSL 1.1.0 或更高版本 (除非 `./configure --disable-tls` 已停用 TLS 支援)。
- SASL (libsasl2 , 除非 `./configure --disable-sasl` 已停用 SASL 支援)。

### 編譯 aws-elasticache-cluster-client-libmemcached 程式庫

1. 啟動 Amazon EC2 執行個體。
2. 安裝程式庫的依存項目。
  - 在 Amazon Linux 201509 AMI/Amazon Linux 2 AMI 上

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

- 在 Ubuntu 14.04 AMI 上 (OpenSSL >= 1.1 的 Ubuntu 版本不需要)

```
sudo apt-get update
```

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

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

### 3. 提取儲存庫並編譯程式碼。

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

如果執行 `../configure` 找不到 `libssl` (OpenSSL 程式庫)，可能需要調整 `PKG_CONFIG_PATH` 環境變數：

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

或者，如果您未使用 TLS，可以執行以下命令加以停用：

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

編譯適用於 PHP 的 ElastiCache Memcached 自動探索用戶端

以下各節說明如何編譯 ElastiCache Memcached 自動探索用戶端

主題

- [編譯適用於 PHP 7 或更高版本的 ElastiCache Memcached 用戶端](#)
- [編譯適用於 PHP 5 的 ElastiCache Memcached 用戶端](#)

編譯適用於 PHP 7 或更高版本的 ElastiCache Memcached 用戶端

以您目前使用的版本取代 PHP-7.x。

安裝 PHP：

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

在程式碼目錄下執行以下命令組。

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

如果執行 `../configure` 找不到 `libssl` (OpenSSL 程式庫)，可能需要將 `PKG_CONFIG_PATH` 環境變數調整為 OpenSSL 的 `.PC` 檔案目錄：

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

或者，如果您未使用 TLS，可以執行以下命令加以停用：

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

### Note

您可以將 `libmemcached` 程式庫靜態連結至 PHP 二進位檔，讓它可以在各種 Linux 平台間移植。若要執行此作業，請在 `make` 前執行以下命令：

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

## 編譯適用於 PHP 5 的 ElastiCache Memcached 用戶端

透過在 `aws-elasticache-cluster-client-memcached-for-php/` 資料夾下執行以下命令，來編譯 `aws-elasticache-cluster-client-memcached-for-php`。

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

## 設定 ElastiCache 用戶端

ElastiCache 叢集具 Memcached 通訊協定合規性。程式碼、應用程式，以及大多數您目前搭配現有 Memcached 環境使用的熱門工具都可順暢地與服務搭配使用。

本節討論連線到 ElastiCache 中快取節點的具體考量事項。

### 主題

- [尋找節點端點及連接埠號碼](#)
- [進行連線以使用自動探索](#)
- [DNS 名稱與基礎 IP](#)

### 尋找節點端點及連接埠號碼

若要連線到快取節點，您的應用程式需要知道該節點的端點及連接埠號碼。

#### 尋找節點端點及連接埠號碼 (主控台)

#### 判斷節點端點和連接埠號碼

1. 登入 [Amazon ElastiCache 管理主控台](#)，並選擇在您叢集上執行的引擎。  
接著會出現執行所選擇引擎的所有叢集清單。
2. 在您執行的引擎和組態下繼續。
3. 選擇您有興趣的叢集名稱。
4. 尋找您有興趣節點的 Port (連接埠) 及 Endpoint (端點) 欄。

#### 尋找快取節點端點及連接埠號碼 (AWS CLI)

若要判斷快取節點端點和連接埠號碼，請搭配 `--show-cache-node-info` 參數使用 `describe-cache-clusters` 命令。

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

完整 DNS 名稱和連接埠號碼會位於輸出的 Endpoint (端點) 區段。

#### 尋找快取節點端點及連接埠號碼 (ElastiCache API)

若要判斷快取節點端點和連接埠號碼，請搭配 `ShowCacheNodeInfo=true` 參數使用 `DescribeCacheClusters` 動作。

## Example

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

### 進行連線以使用自動探索

若您的應用程式使用自動探索，您只需要知道叢集的組態端點，而無須知道每個快取節點的個別端點。如需更多詳細資訊，請參閱 [自動識別叢集中的節點](#)。

#### Note

目前，自動探索僅供執行 Memcached 的快取叢集使用。

### DNS 名稱與基礎 IP

用戶端會維持一份伺服器清單，其中包含保有快取資料的伺服器地址及連接埠。使用 ElastiCache 時，DescribeCacheClusters API (或 describe-cache-clusters 命令列公用程式) 會傳回可用於伺服器清單的完整 DNS 項目和連接埠號碼。

#### Important

將用戶端應用程式設為在嘗試連線到快取節點端點時頻繁解析快取節點的 DNS 名稱很重要。

### VPC 安裝

ElastiCache 會在快取節點從故障還原時，確認快取節點的 DNS 名稱和 IP 地址都維持相同。

## 非 VPC 安裝

ElastiCache 會在快取節點從故障還原時，確認快取節點的 DNS 名稱沒有變更，但是快取節點的基礎 IP 地址可以變更。

根據預設，大多數的用戶端程式庫都支援持久性快取節點連線。建議使用 ElastiCache 時使用持久性快取節點連線。用戶端的 DNS 快取可能會在多個位置發生，包括用戶端程式庫、語言執行時間，或是用戶端作業系統。建議您檢閱每一個 layer 的應用程式組態，確保為您的快取節點頻繁解析 IP 地址。

## 準備叢集

以下說明如何使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 來建立叢集。

您也可以使用 [AWS CloudFormation](#) 建立 ElastiCache 叢集。如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的 [AWS::ElastiCache::CacheCluster](#)，其中包括如何實作該做法的指引。

每當您建立叢集，建議您進行一些準備工作，將來就不需要升級或當下進行變更。

### 主題

- [判斷要求](#)
- [選擇您的節點大小](#)

## 判斷要求

### 準備

釐清下列問題的答案有助於提高叢集的建立流暢度：

您要使用 ElastiCache 無伺服器或執行個體型服務嗎？

如果您想要使用無伺服器快取，只要確定已正確設定 VPC、子網路和安全群組即可。如需詳細資訊，請參閱在 [Amazon VPC 中存取 ElastiCache 快取的存取模式](#)。如果您想使用基於實例的 ElastiCache，請繼續閱讀。

- 您需要哪種類型的節點執行個體？

如需選擇執行個體節點類型的指導方針，請參閱[選擇您的 Memcached 節點大小](#)。

- 您是否會在以 Amazon VPC 為基礎的 Virtual Private Cloud (VPC) 中啟動叢集？

### ⚠ Important

如果您要在 VPC 中啟動叢集，請先在相同 VPC 中建立子網路群組，再開始建立叢集。如需詳細資訊，請參閱 [子網路和子網路群組](#)。

ElastiCache 旨在 AWS 使用 Amazon EC2 從內部訪問。不過，如果您在以 Amazon VPC 為基礎的 VPC 中啟動叢集，並且叢集在 VPC 中，則可以從 AWS 外部提供存取。如需詳細資訊，請參閱 [在 Amazon VPC 中存取 ElastiCache 快取的存取模式](#)。

- 您是否需要自訂任何參數值？

如果需要，請建立自訂參數群組。如需詳細資訊，請參閱 [建立參數群組](#)。

- 您是否需要建立自己的 VPC 安全群組？

如需詳細資訊，請參閱 [VPC 中的安全性](#)。

- 您要如何實作容錯能力？

如需詳細資訊，請參閱 [減少故障](#)。

## 主題

- [記憶體和處理器要求](#)
- [Memcached 叢集組態](#)
- [擴展要求](#)
- [存取要求](#)
- [區域、可用區域與本地區域要求](#)

## 記憶體和處理器要求

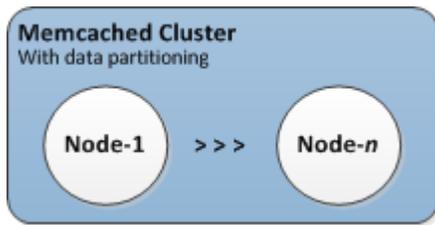
Amazon 的基本構建塊 ElastiCache 是節點。您可以逐一設定節點或以群組形式設定節點，以形成叢集。當您判斷要為叢集使用何種節點類型時，請一併考量叢集的節點組態和您要存放的資料量。

Memcached 是多執行緒的引擎，因此節點的核心數量會影響叢集可用的運算能力。

## Memcached 叢集組態

ElastiCache 對於內存緩存集群由 1 到 60 個節點組成。Memcached 叢集中的資料會分割到叢集的各節點中。您的應用程式會使用網路位址 (稱為「端點」) 來與 Memcached 叢集連線。Memcached 叢集

中的每個節點都有自己的端點，您的應用程式會使用這些端點來讀取或寫入特定節點。除了節點端點之外，Memcached 叢集本身還有一個端點名為組態端點。您的應用程式可使用此端點來讀取或寫入叢集，並由 [自動識別叢集中的節點](#) 判斷要讀取或寫入哪些節點。



如需詳細資訊，請參閱 [管理叢集](#)。

### 擴展要求

您可以使用更大的新節點類型來建立新叢集，以擴展所有叢集的規模。當你擴展 Memcached 叢集時，新叢集會開始為空。

Amazon ElastiCache 的 Memcached 叢集可以向外擴展或縮減。若要擴展或縮減 Memcached 叢集的規模，您只要新增或移除叢集的節點即可。如果您已啟用 Automatic Discovery 且應用程式已連線至叢集的組態端點，則當您新增或移除節點時不需要對應用程式進行任何變更。

如需詳細資訊，請參閱本指南中的 [縮放記憶 ElastiCache 體快取](#)。

### 存取要求

根據設計，可以從 Amazon EC2 執行個體存取亞馬遜 ElastiCache 叢集。對 ElastiCache 叢集的網路存取僅限於建立叢集的帳戶。因此，在您從 Amazon EC2 執行個體存取叢集之前，必須先授權讓 Amazon EC2 執行個體存取叢集。執行此作業的步驟依據您要在 EC2-VPC 或 EC2-Classic 中啟動而異。

如果您是在 EC2-VPC 中啟動叢集，則需要授權網路輸入至您的叢集。如果您將叢集啟動到 EC2-Classic，則需要將與執行個體存取權關聯的 Amazon 彈性運算雲端安全群組授與 ElastiCache 安全群組。如需詳細說明，請參閱本指南的 [存取您的叢集](#)。

### 區域、可用區域與本地區域要求

Amazon ElastiCache 支持所有 AWS 區域。透過將 ElastiCache 叢集定位在應用程式附近的 AWS 區域中，您可以減少延遲。如果您的叢集有多個節點，將節點安置在不同可用區域或 Local Zones 中可降低故障對叢集的影響。

如需詳細資訊，請參閱下列內容：

- [區域與可用區域](#)
- [本地區域](#)
- [減少故障](#)

## 選擇您的節點大小

您為叢集選取的節點大小會影響成本、效能及容錯能力。

### 選擇您的 Memcached 節點大小

Memcached 叢集包含一或多個節點，其中叢集的資料會分割在各個節點中。因此，叢集的記憶體需求與節點的記憶體相關，但不相同。您可以透過擁有幾個大型節點或數個較小的節點，來達到您需要的叢集記憶體容量。除此之外，隨著您的需求變更，您可以新增節點或從叢集移除節點，來只為您需要的項目支付費用。

您叢集的總記憶體容量是透過將叢集中的節點數與每個節點的 RAM 容量相乘，再減去系統額外負荷計算而得。每個節點的容量是根據節點類型而定。

$$\text{cluster\_capacity} = \text{number\_of\_nodes} * (\text{node\_capacity} - \text{system\_overhead})$$

叢集中的節點數量是您執行 Memcached 叢集可用性的關鍵因素。單一節點發生故障，可能會影響您應用程式的可用性及後端資料庫的負載。在這種情況下，ElastiCache 會為失敗的節點佈建一個替代節點，並重新填入資料。您可以將記憶體和運算容量分散到更多容量較小的節點間，而非使用少數具備高容量的節點，藉此降低此可用性影響。

在希望擁有 35 GB 快取記憶體的案例中，可以透過以下任一組態進行設定：

- 11 個 `cache.t2.medium` 節點，其具備 3.22 GB 的記憶體及 2 個執行緒，合計 35.42 GB 和 22 個執行緒。
- 6 個 `cache.m4.large` 節點，其具備 6.42 GB 的記憶體及 2 個執行緒，合計 38.52 GB 和 12 個執行緒。
- 3 個 `cache.r4.large` 節點，其具備 12.3 GB 的記憶體及 2 個執行緒，合計 36.90 GB 和 6 個執行緒。
- 3 個 `cache.m4.xlarge` 節點，其具備 14.28 GB 的記憶體及 4 個執行緒，合計 42.84 GB 和 12 個執行緒。

## 比較節點選項

節點類型	記憶體 (以 GiB 為單位)	核心	每小時成 本 *	所需節點 數	總記憶體 (以 GiB 為單位)	總核心數	每月成 本 /
cache.t2. medium	3.22	2	0.068 USD	11	35.42	22	538.56 USD
cache.m4. large	6.42	2	0.156 USD	6	38.52	12	673.92 USD
cache.m4. xlarge	14.28	4	0.311 USD	3	42.84	12	671.76 USD
cache.m5. xlarge	12.93	4	0.311 USD	3	38.81	12	671.76 USD
cache.m6g .large	6.85	2	\$ 0.147	6	41.1	12	\$ 635
cache.r4. large	12.3	2	0.228 USD	3	36.9	6	492.48 USD
cache.r5. large	13.07	2	0.216 USD	3	39.22	6	466.56 USD
cache.r6g .large	13.07	2	\$ 0.205	3	42.12	6	\$ 442

\* 截至 2020 年 10 月 8 日的每個節點小時成本。

用量 100%，持續 30 天 (720 小時) 的每月費用。

這些選項每個都能提供相似的記憶體容量，但不同的運算容量及成本。若要比較您具體選項的成本，請參閱 [Amazon ElastiCache 定價](#)。

針對執行 Memcached 的叢集，每個節點上一部分的可用記憶體會用於連線額外負荷。如需詳細資訊，請參閱 [Memcached 連線額外負荷](#)

使用多個節點需要將索引鍵分散到這些節點當中。每個節點都有自己的端點。為了簡化端點管理，您可以使用 ElastiCache 的自動探索功能，讓用戶端程式能自動識別叢集中的所有節點。如需更多詳細資訊，請參閱 [自動識別叢集中的節點](#)。

在某些案例中，您可能不確定需要多少容量。這個情況下，建議從一個 `cache.m5.large` 節點開始測試。然後使用發佈到 Amazon CloudWatch 的 ElastiCache 指標，監控記憶體用量、CPU 使用率及快取命中率。如需有關 ElastiCache 的 CloudWatch 指標的詳細資訊，請參閱「[使用 CloudWatch 指標監控用量](#)」。針對生產及更大的工作負載，R5 節點可提供最佳效能及 RAM 成本值。

若您的叢集沒有達到所需的命中率，您可輕易新增更多節點來增加叢集中可用的總記憶體。

若您的叢集具備足夠的命中率，但受到 CPU 限制，請使用提供更強運算能力的節點類型設定新的叢集。

## 建立叢集

下列範例說明如何使用 AWS Management Console、AWS CLI 和 ElastiCache API 建立叢集。

### 建立 Memcached 叢集 (主控台)

當您使用 Memcached 引擎時，Amazon ElastiCache 支持在多個節點上水平分區數據。Memcached 可啟用自動探索功能，因此您不需要追蹤每個節點的端點。Memcached 可追蹤每個節點的端點，並隨著節點新增與移除節點而更新端點清單。您的應用程式只需要組態端點即可與叢集互動。如需自動探索的詳細資訊，請參閱[自動識別叢集中的節點](#)。

若要建立 Memcached 叢集，請按照 [步驟 1：建立快取](#) 的步驟執行

一旦叢集的狀態變為可用，您就可以為其授予 Amazon EC2 存取權限、連線至叢集並開始使用叢集。如需更多詳細資訊，請參閱「[存取您的叢集](#)」及「[手動連線至快取節點](#)」。

#### Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱[刪除叢集](#)。

### 建立叢集 (AWS CLI)

若要使用建立叢集 AWS CLI，請使用 `create-cache-cluster` 指令。

#### Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未主動使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱[刪除叢集](#)。

### 建立 Memcached 快取叢集 (AWS CLI)

下列 CLI 程式碼會建立含 3 個節點的 Memcached 快取叢集。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-cluster \  
--cache-cluster-id my-cluster \  

```

```
--cache-node-type cache.r4.large \  
--engine memcached \  
--engine-version 1.4.24 \  
--cache-parameter-group default.memcached1.4 \  
--num-cache-nodes 3
```

針對 Windows :

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

## 建立叢集 (ElastiCache API)

若要使用 ElastiCache API 建立叢集，請使用 `CreateCacheCluster` 動作。

### Important

在您的叢集可用之後，系統就會按叢集作用中時間每個小時或部分小時計費 (即使您並未使用亦同)。若要停止此叢集產生費用，您必須將其刪除。請參閱 [刪除叢集](#)。

## 創建一個內存緩存集群 (ElastiCache API)

下面的代碼創建具有 3 個節點 (ElastiCache API) 的內存緩存集群。

加上分行符號的用意是便於閱讀。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=CreateCacheCluster  
&CacheClusterId=my-cluster  
&CacheNodeType=cache.r4.large  
&Engine=memcached  
&NumCacheNodes=3  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150508T220302Z
```

```
&Version=2015-02-02
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=<credential>
&X-Amz-Date=20150508T220302Z
&X-Amz-Expires=20150508T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Signature=<signature>
```

## 檢視叢集的詳細資訊

您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 檢視一或多個叢集的詳細資訊。

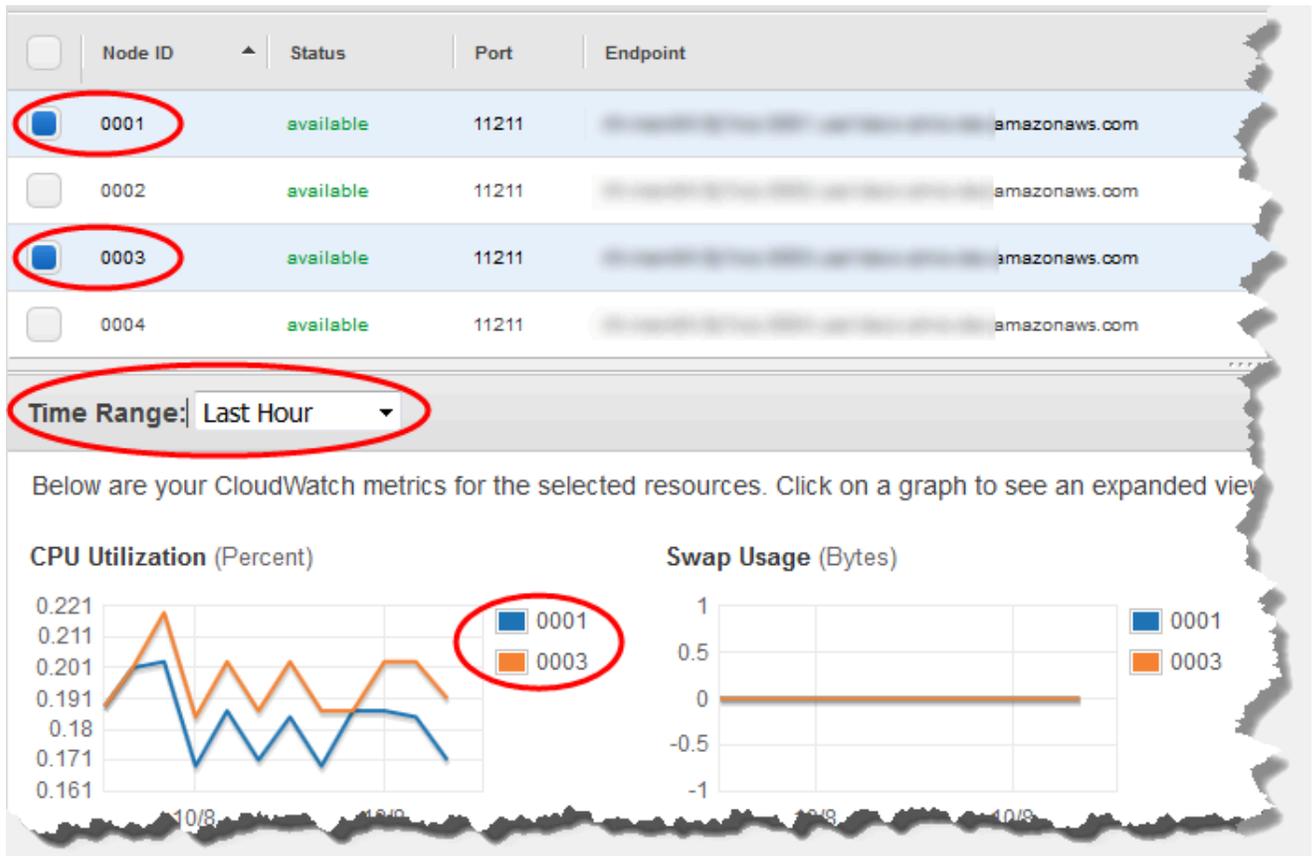
### 檢視叢集的詳細資訊 (主控台)

您可以使用 ElastiCache 主控台、AWS CLI for ElastiCache 或 ElastiCache API 來檢視 Memcached 叢集的詳細資訊。

下列程序詳細說明如何使用 ElastiCache 主控台來檢視 Memcached 叢集的詳細資訊。

### 檢視 Memcached 叢集的詳細資訊

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 Amazon ElastiCache 主控台。
2. 從右上角的清單中選擇所需的 AWS 區域。
3. 在 ElastiCache 主控台儀表板中，選擇 Memcached。這麼做會列出正在執行任何版本之 Memcached 的所有叢集。
4. 若要查看叢集的詳細資訊，請選擇叢集名稱左側的方塊。
5. 檢視節點資訊：
  - a. 選擇叢集名稱。
  - b. 選擇 Nodes (節點) 標籤。
  - c. 若要檢視一或多個節點的指標，請選擇節點 ID 左側的方塊，接著從 Time range (時間範圍) 清單選擇指標的時間範圍。選擇多個節點會產生重疊圖形。



兩個 Memcached 節點在過去一小時的指標

### 檢視叢集的詳細資訊 (AWS CLI)

您可以使用 AWS CLI `describe-cache-clusters` 命令來檢視叢集的詳細資訊。如果省略 `--cache-cluster-id` 參數，則會傳回多個叢集 (最多 `--max-items` 個) 的詳細資訊。如果包含 `--cache-cluster-id` 參數，則會傳回指定叢集的詳細資訊。您可以使用 `--max-items` 參數限制傳回的記錄數量。

以下程式碼會列出 `my-cluster` 的詳細資訊。

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

以下程式碼清單會列出最多 25 個叢集的詳細資訊。

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

## Example

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

此作業會產生類似下列的輸出 (JSON 格式)：

```
{  
  "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"
```

```
    }  
  ]  
}
```

如需詳細資訊，請參閱 ElastiCache 的 AWS CLI 主題 [describe-cache-clusters](#)。

### 檢視叢集的詳細資訊 (ElastiCache API)

您可以使用 ElastiCache API DescribeCacheClusters 動作來檢視叢集的詳細資訊。如果包含 CacheClusterId 參數，則會傳回指定叢集的詳細資訊。如果省略 CacheClusterId 參數，則會傳回最多 MaxRecords 個叢集 (預設值為 100) 的詳細資訊。MaxRecords 的值不可小於 20 或大於 100。

以下程式碼會列出 my-cluster 的詳細資訊。

```
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>
```

以下程式碼清單會列出最多 25 個叢集的詳細資訊。

```
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>
```

如需詳細資訊，請參閱 ElastiCache API 參考主題 [DescribeCacheClusters](#)。

## 修改 ElastiCache 叢集

除了新增或移除叢集的節點之外，有時候您可能需要對現有的叢集做其他變更，例如新增安全群組、變更維護時段或參數群組。

建議您將維護時段落在使用量最低的時段。您可能需要不時進行調整。

當變更叢集的參數時，變更即會立即套用至叢集，或在叢集重新啟動之後套用。無論是變更叢集的參數群組本身或是叢集的參數群組內的參數值，均適用此情況。若要決定何時套用特定的參數變更，請參閱 [Memcached 專用參數](#) 和 資料表中的 Changes Take Effect (變更生效) 欄位。如需重新啟動叢集的資訊，請參閱 [重新啟動叢集](#)。

使用 AWS Management Console

### 修改叢集

1. 請登入 AWS Management Console 並開啟 ElastiCache 主控台，網址為 <https://console.aws.amazon.com/elasticache/>。
2. 從右上角的清單中，選擇您要修改之叢集所在的 AWS 區域。
3. 在導覽窗格中，選擇您要修改之叢集所執行的引擎。

接著會出現所選引擎的叢集清單。

4. 在叢集清單中，針對您要修改的叢集，選擇其名稱。
5. 選擇 Actions (動作)，然後選擇 Modify (修改)。

Modify Cluster (修改叢集) 視窗隨即出現。

6. 在 Modify Cluster (修改叢集) 視窗中，進行所需的修改。選項包括：
  - 引擎版本相容性
  - VPC 安全群組
  - 參數群組
  - 維護時段
  - SNS 通知的主題

Apply Immediately (立即套用) 方塊僅適用於引擎版本的修改。若要立即套用變更，請選擇 Apply Immediately (立即套用) 核取方塊。如果未選擇此方塊，引擎版本的修改將於下次維護時段套用。其他修改 (例如變更維護時段) 則會立即套用。

## 7. 選擇 Modify (修改)。

### 使用 AWS CLI

您可以使用 AWS CLI `modify-cache-cluster` 作業修改現有叢集。若要修改叢集的組態值，請指定叢集 ID、要變更的參數以及參數的新值。下方範例會變更名稱為 `my-cluster` 之叢集的維護時段，並立即套用變更。

#### Important

您可以升級到較新的引擎版本。如需這項作業的詳細資訊，請參閱[引擎版本和升級](#)。不過，您無法降級至舊版引擎，除非您刪除現有的叢集，並重新建立叢集。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

`--apply-immediately` 參數僅適用於修改引擎版本和變更叢集節點數。如果您要立即套用其中任何變更，請使用 `--apply-immediately` 參數。如果您想將這些變更延期至下一個維護時段，請使用 `--no-apply-immediately` 參數。其他修改 (例如變更維護時段) 則會立即套用。

如需詳細資訊，請參閱「AWS CLI 的」ElastiCache 主題[modify-cache-cluster](#)。

### 應用 ElastiCache 程式介面

您可以使用 ElastiCache API `ModifyCacheCluster` 作業修改現有叢集。若要修改叢集的組態值，請指定叢集 ID、要變更的參數以及參數的新值。下方範例會變更名稱為 `my-cluster` 之叢集的維護時段，並立即套用變更。

**⚠ Important**

您可以升級到較新的引擎版本。如需這項作業的詳細資訊，請參閱[引擎版本和升級](#)。不過，您無法降級至舊版引擎，除非您刪除現有的叢集，並重新建立叢集。

加上分行符號的用意是便於閱讀。

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=ModifyCacheCluster  
  &CacheClusterId=my-cluster  
  &PreferredMaintenanceWindow=sun:23:00-mon:02:00  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20150901T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20150202T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20150901T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

`ApplyImmediately` 參數僅適用於節點類型、引擎版本和叢集節點數的修改。如果您要立即套用其中任何變更，請將 `ApplyImmediately` 參數設為 `true`。如果您想將這些變更延期至下一個維護時段，請將 `ApplyImmediately` 參數設為 `false`。其他修改 (例如變更維護時段) 則會立即套用。

如需詳細資訊，請參閱 ElastiCache API 參考主題[ModifyCacheCluster](#)。

## 重新啟動叢集

有些變更需要重新啟動叢集才能套用變更。例如，針對某些參數，當您變更參數群組中的參數值時，只會在重新啟動時套用。

當您重新啟動叢集時，叢集會排清其所有資料並重新啟動引擎。程序進行期間，您無法存取叢集。由於叢集會排清其所有資料，因此當叢集再次可用時，就是一個空的叢集供您使用。

您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 重新啟動叢集。不論您使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API，都只能重新啟動單一叢集。若要重新啟動多個叢集，您必須反覆執行程序和操作。

### 使用 AWS Management Console

您可以使用 ElastiCache 主控台重新啟動叢集。

#### 重新啟動叢集 (主控台)

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 從右上角的清單中選擇所需的 AWS 區域。
3. 在導覽窗格中，選擇您要重新啟動之叢集所執行的引擎。

接著會出現執行所選引擎的叢集清單。

4. 選擇叢集名稱左側的按鈕，以選擇要重新啟動的叢集。

選擇動作，然後選擇重新啟動。

如果您選擇一個以上的叢集，Reboot (重新啟動) 按鈕則不會啟用。

若要重新啟動多個叢集，請針對每個要重新啟動的叢集重複步驟 2 到 5。您不需要等待某個叢集完成重新啟動，即可重新啟動其他叢集。

若要重新啟動特定節點，請選擇節點，然後選擇 Reboot (重新啟動)。

### 使用 AWS CLI

若要重新啟動叢集 (AWS CLI)，請使用 `reboot-cache-cluster` CLI 作業。

若要重新啟動叢集中的特定節點，請使用 `--cache-node-ids-to-reboot` 列出要重新啟動的特定叢集。下列命令會重新啟動 `my-cluster` 的 0001、0002 和 0004 節點。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

若要重新啟動叢集中的所有節點，請使用 `--cache-node-ids-to-reboot` 參數並列出所有叢集的節點 ID。如需詳細資訊，請參閱 [reboot-cache-cluster](#)。

使用 ElastiCache API

若要使用 ElastiCache API 重新啟動叢集，請使用 `RebootCacheCluster` 動作。

若要重新啟動叢集中的特定節點，請使用 `CacheNodeIdsToReboot` 列出要重新啟動的特定叢集。下列命令會重新啟動 `my-cluster` 的 0001、0002 和 0004 節點。

```
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>
```

若要重新啟動叢集中的所有節點，請使用 `CacheNodeIdsToReboot` 參數並列出所有叢集的節點 ID。如需詳細資訊，請參閱 [RebootCacheCluster](#)。

## 將節點新增至叢集

將節點新增至 Memcached 叢集時，會增加叢集的分割區數。當您變更叢集中的分割區數量時，您必須重新對應一些鍵空間，以讓它們對應到正確的節點。重新對應鍵空間時，會暫時增加叢集的快取未命中數。如需更多詳細資訊，請參閱 [設定您的 ElastiCache 用戶端以達到高效率的負載平衡](#)。

您可以使用 ElastiCache 管理主控台、AWS CLI 或 ElastiCache API 來將節點新增至您的叢集。

使用 AWS Management Console

主題

- [將節點新增至叢集 \(主控台\)](#)

將節點新增至叢集 (主控台)

下列程序可用來將節點新增至叢集。

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇您要為其新增節點之叢集所執行的引擎。

接著會出現執行所選引擎的叢集清單。

3. 從叢集清單中，針對您要新增節點的叢集，選擇其名稱。
4. 選擇 Add node (新增節點)。
5. 完成 Add Node (新增節點) 對話方塊中請求的資訊。
6. 選擇 Apply Immediately - Yes (立即套用 - 是) 按鈕可立即新增此節點；或選擇 No (否)，在此叢集的下一個維護時段期間，新增此節點。

新增和移除要求對特定請求的影響

案例	待定作業	新請求	結果
案例 1	Delete	Delete	新的刪除請求 (待定或立即) 會取代待定的刪除請求。  例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出刪除節點 0002 和 0004 的新請求，則只會刪除節點 0002 和 0004。系統不會刪除節點 0001、0003 和 0007。
案例 2	Delete	建立	新的建立請求 (待定或立即) 會取代待定的刪除請求。  例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出建立節點的新請求，則會建立新的節點，而不會刪除節點 0001、0003 和 0007。

案例	待定作業	新請求	結果
案例 3	建立	Delete	<p>新的刪除請求 (待定或立即) 會取代待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出刪除節點 0003 的新請求，則不會建立新的節點，且會刪除節點 0003。</p>
案例 4	建立	建立	<p>新的建立請求會新增至待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出建立三個節點的新請求，則新請求會新增至待定請求，並建立五個節點。</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>⚠ Important</b></p> <p>如果新的建立請求設定為 Apply Immediately - Yes (立即套用 - 是)，則會立即執行所有建立請求。如果新的建立請求設定為 Apply Immediately - No (立即套用 - 否)，則所有建立請求都處於待定狀態。</p> </div>

若要判斷哪些作業處於待定狀態，請選擇 Description (描述) 索引標籤，並檢查顯示多少待定的建立或刪除項目。待定建立和待定刪除項目不能同時存在。

#### 7. 選擇 Add (新增) 按鈕。

在一段時間之後，新的節點應會顯示在節點清單中，且狀態為 creating (建立中)。如果並未顯示，請重新整理瀏覽器頁面。如果節點狀態變更為「可用」，新的節點即可供使用。

#### 使用 AWS CLI

若要使用 AWS CLI 將節點新增至叢集，請使用 AWS CLI 操作 `modify-cache-cluster` 搭配下列參數：

- `--cache-cluster-id` 您想為其新增節點的快取叢集 ID。
- `--num-cache-nodes` `--num-cache-nodes` 參數可指定套用修改之後您要在此叢集中使用的節點數。若要新增節點至此叢集，`--num-cache-nodes` 必須大於此叢集中的目前節點數。如果此值小

於目前節點數，ElastiCache 會預期收到 `cache-node-ids-to-remove` 參數以及要從叢集中移除的節點清單。如需更多詳細資訊，請參閱 [使用 AWS CLI](#)。

- `--apply-immediately` 或 `--no-apply-immediately` 指定是否要立即新增這些節點，或等到下個維護時段再新增。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

此作業會產生類似下列的輸出 (JSON 格式)：

```
{  
  "CacheCluster": {  
    "Engine": "memcached",  
    "CacheParameterGroup": {  
      "CacheNodeIdsToReboot": [],  
      "CacheParameterGroupName": "default.memcached1.4",  
      "ParameterApplyStatus": "in-sync"  
    },  
    "CacheClusterId": "my-cluster",  
    "PreferredAvailabilityZone": "us-west-2b",  
    "ConfigurationEndpoint": {  
      "Port": 11211,  
      "Address": "r1lh-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",  
  }  
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [modify-cache-cluster](#)。

## 使用 ElastiCache API

### 將節點新增至叢集 (ElastiCache API)

- 使用下列參數呼叫 ModifyCacheCluster API 操作：
  - CacheClusterId 您想為其新增節點的叢集 ID。
  - NumCacheNodes NumCachNodes 參數可指定套用修改之後您要在此叢集中使用的節點數。若要新增節點至此叢集，NumCacheNodes 必須大於此叢集中的目前節點數。如果此值小於目前節點數，ElastiCache 會預期收到 CacheNodeIdsToRemove 參數以及要從叢集中移除的節點清單 (請參閱「[使用 ElastiCache API](#)」)。
  - ApplyImmediately 指定是否要立即新增這些節點，或等到下個維護時段再新增。
  - Region 指定您想為其新增節點之叢集的 AWS 區域。

下列範例說明可將節點新增至叢集的呼叫。

### Example

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster
```

```
&ApplyImmediately=true
&NumCacheNodes=5
&CacheClusterId=my-cluster
&Region=us-east-2
&Version=2014-12-01
&SignatureVersion=4
&SignatureMethod=HmacSHA256
&Timestamp=20141201T220302Z
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 ElastiCache API 主題 [ModifyCacheCluster](#)。

## 從叢集中移除節點

每當您變更 Memcached 叢集中的節點數量時，必須至少重新對應一些索引鍵空間，使其對應到正確的節點。如需有關負載平衡 Memcached 叢集的詳細資訊，請參閱[設定您的 ElastiCache 用戶端以達到高效率的負載平衡](#)。

您可以使用 AWS Management Console、AWS CLI 或 ElastiCache API 從叢集中移除節點。

### 使用 AWS Management Console

#### 從叢集中移除節點 (主控台)

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在右上角清單中，選擇您要從中移除節點之叢集的 AWS 區域。
3. 在導覽窗格中，選擇您要從中移除節點之叢集所執行的引擎。

接著會出現執行所選引擎的叢集清單。

4. 在叢集清單中，選擇您要從中移除節點的叢集名稱。

接著會出現叢集的節點清單。

5. 選擇您要移除之節點的節點 ID 旁的方塊。使用 ElastiCache 主控台時，您一次只能刪除一個節點，因此選擇多個節點表示您無法使用 Delete node (刪除節點) 按鈕。

Delete Node (刪除節點) 頁面隨即出現。

6. 若要刪除節點，請完成 Delete Node (刪除節點) 頁面，然後選擇 Delete Node (刪除節點)。如要保留節點，請選擇 Cancel (取消)。

#### 新增和移除請求對特定請求的影響

案例	待定作業	新請求	結果
案例 1	Delete	Delete	新的刪除請求 (待定或立即) 會取代待定的刪除請求。  例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出刪除節點 0002 和 0004 的新請求，則只會刪除節點 0002 和 0004。系統不會刪除節點 0001、0003 和 0007。

案例	待定作業	新請求	結果
案例 2	Delete	建立	<p>新的建立請求 (待定或立即) 會取代待定的刪除請求。</p> <p>例如，如果節點 0001、0003 和 0007 處於待定刪除狀態，並發出建立節點的新請求，則會建立新的節點，而不會刪除節點 0001、0003 和 0007。</p>
案例 3	建立	Delete	<p>新的刪除請求 (待定或立即) 會取代待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出刪除節點 0003 的新請求，則不會建立新的節點，且會刪除節點 0003。</p>
案例 4	建立	建立	<p>新的建立請求會新增至待定的建立請求。</p> <p>例如，若存在建立兩個節點的待定請求，並發出建立三個節點的新請求，則新請求會新增至待定請求，並建立五個節點。</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p><b>⚠ Important</b></p> <p>如果新的建立請求設定為 Apply Immediately - Yes (立即套用 - 是)，則會立即執行所有建立請求。如果新的建立請求設定為 Apply Immediately - No (立即套用 - 否)，則所有建立請求都處於待定狀態。</p> </div>

若要判斷哪些作業處於待定狀態，請選擇 Description (描述) 索引標籤，並檢查顯示多少待定的建立或刪除項目。待定建立和待定刪除項目不能同時存在。

## 使用 AWS CLI

1. 找出您要移除的節點 ID。如需更多詳細資訊，請參閱 [檢視叢集的詳細資訊](#)。
2. 搭配使用 `modify-cache-cluster` CLI 操作與要移除的節點清單，如下列範例所示。

若要使用命令列界面移除叢集中的節點，請搭配使用 `modify-cache-cluster` 命令與下列參數：

- `--cache-cluster-id` 您想要從中移除節點的快取叢集 ID。
- `--num-cache-nodes` `--num-cache-nodes` 參數可指定套用修改之後您要在此叢集中使用的節點數。
- `--cache-node-ids-to-remove` 您要從這個叢集中移除的節點 ID 清單。
- `--apply-immediately` 或 `--no-apply-immediately` 指定是否要立即移除這些節點，或等到下個維護時段再移除。
- `--region` 指定您想要從中移除節點之叢集的 AWS 區域。

下列範例會從 `my-cluster` 叢集中立即移除節點 `0001`。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

此作業會產生類似下列的輸出 (JSON 格式)：

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

```
"CacheClusterId": "my-cluster",
"PreferredAvailabilityZone": "us-east-2b",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "r1h-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",
}
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [describe-cache-cluster](#) 和 [modify-cache-cluster](#)。

## 使用 ElastiCache API

若要使用 ElastiCache API 移除節點，請搭配使用快取叢集 ID 與要移除的節點清單來呼叫 `ModifyCacheCluster` API 作業，如下所示：

- `CacheClusterId` 您想要從中移除節點的快取叢集 ID。
- `NumCacheNodes` `NumCacheNodes` 參數可指定套用修改之後您要在此叢集中使用的節點數。

- `CacheNodeIdsToRemove.member.n` 要從叢集中移除的節點 ID 清單。
  - `CacheNodeIdsToRemove.member.1=0004`
  - `CacheNodeIdsToRemove.member.1=0005`
- `ApplyImmediately` 指定是否要立即移除這些節點，或等到下個維護時段再移除。
- `Region` 指定您想要從中移除節點之叢集的 AWS 區域。

下列範例會從 my-cluster 叢集中立即移除節點 0004 和 0005。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=ModifyCacheCluster  
&CacheClusterId=my-cluster  
&ApplyImmediately=true  
&CacheNodeIdsToRemove.member.1=0004  
&CacheNodeIdsToRemove.member.2=0005  
&NumCacheNodes=3  
&Region us-east-2  
&Version=2014-12-01  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20141201T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20141201T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20141201T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 ElastiCache API 主題 [ModifyCacheCluster](#)。

## 取消待定的新增或刪除節點作業

如果您選擇不要立即套用變更，則操作會具備 pending (等待中) 狀態，直到其於下一個維護時段執行為止。您可以取消任何等待中的操作。

### 取消等待中的操作

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在右上角清單中，選擇您要取消待定新增或刪除節點作業的 AWS 區域。
3. 在導覽窗格中，選擇含有您要取消之等待中操作的叢集所執行的引擎。接著會出現執行所選引擎的叢集清單。
4. 在叢集清單中，選擇含有您要取消之等待中操作的叢集名稱，而非叢集名稱左側的方塊。
5. 若要判斷哪些作業處於待定狀態，請選擇 Description (描述) 索引標籤，並檢查顯示多少待定的建立或刪除項目。待定建立和待定刪除項目不能同時存在。
6. 選擇 Nodes (節點) 標籤。
7. 若要取消所有等待中的操作，請選擇 Cancel Pending (取消待定)。隨即顯示 Cancel Pending (取消待定) 對話方塊。
8. 您可以選擇 Cancel Pending (取消待定) 按鈕以取消所有等待中的操作，或選擇 Cancel (取消) 來保留操作。

## 刪除叢集

一旦叢集處於「可用」狀態，就會開始向您收費，不論您是否主動使用亦同。若要停止產生費用，請刪除叢集。

### 使用 AWS Management Console

以下程序會從您的部署中刪除單一叢集。若要刪除多個叢集，請針對每個要刪除的叢集重複此程序。您不需要等待某個叢集完成刪除，即可開始刪除其他叢集。

### 刪除叢集

1. 前往 <https://console.aws.amazon.com/elasticache/>，登入 AWS Management Console 並開啟 Amazon ElastiCache 主控台。
2. 在 ElastiCache 主控台儀表板中，選擇您要刪除之叢集執行的引擎。

接著會出現執行該引擎的所有叢集清單。

3. 要選擇欲刪除的叢集，請從叢集清單選擇叢集的名稱。

#### Important

您一次只能從 ElastiCache 主控台刪除一個叢集。選擇多個叢集會停用刪除操作。

4. 對於 Actions (動作)，請選擇 Delete (刪除)。
5. 在 Delete Cluster (刪除叢集) 確認畫面中，選擇 Delete (刪除) 以刪除叢集，或選擇 Cancel (取消) 來保留叢集。

如果您選擇 Delete (刪除)，叢集的狀態就會變更為 deleting (正在刪除)。

一旦您的叢集不再列於叢集清單，您就不會再因此產生費用。

### 使用 AWS CLI

下列程式碼會刪除 my-cluster 快取叢集。

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

delete-cache-cluster CLI 動作只會刪除一個快取叢集。若要刪除多個快取叢集，請針對每個要刪除的快取叢集呼叫 delete-cache-cluster。您不需要等待某個快取叢集完成刪除，即可刪除其他叢集。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

如需詳細資訊，請參閱 ElastiCache 的 AWS CLI 主題 [delete-cache-cluster](#)。

使用 ElastiCache API

下列程式碼會刪除 `my-cluster` 叢集。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DeleteCacheCluster  
&CacheClusterId=my-cluster  
&Region us-east-2  
&SignatureVersion=4  
&SignatureMethod=HmacSHA256  
&Timestamp=20150202T220302Z  
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
&X-Amz-Date=20150202T220302Z  
&X-Amz-SignedHeaders=Host  
&X-Amz-Expires=20150202T220302Z  
&X-Amz-Credential=<credential>  
&X-Amz-Signature=<signature>
```

DeleteCacheCluster API 操作只會刪除一個快取叢集。若要刪除多個快取叢集，請針對每個要刪除的快取叢集呼叫 DeleteCacheCluster。您不需要等待某個快取叢集完成刪除，即可刪除其他叢集。

如需詳細資訊，請參閱 ElastiCache API 參考主題 [DeleteCacheCluster](#)。

## 存取您的叢集

Amazon ElastiCache 執行個體設計為透過 Amazon EC2 執行個體來存取。

若您在 Amazon Virtual Private Cloud (Amazon VPC) 中啟動 ElastiCache 執行個體，您可以從同一個 Amazon VPC 中的 Amazon EC2 執行個體存取您的 ElastiCache 執行個體。或者，您可以透過使用 VPC 對等互連，從不同 Amazon VPC 中的 Amazon EC2 存取 ElastiCache 執行個體。

如果您在 EC2 Classic 中啟動了 ElastiCache 執行個體，則可透過授權讓與執行個體相關聯的 Amazon EC2 安全群組存取快取安全群組，允許 EC2 執行個體存取您的叢集。根據預設，僅有啟動叢集的帳戶才能存取該叢集。

### 主題

- [授權存取您的叢集](#)

## 授權存取您的叢集

您在 EC2-VPC 中啟動了叢集

如果您將叢集啟動到 Amazon Virtual Private Cloud (Amazon VPC) 中，則只能從在同一個 Amazon VPC 中執行的 Amazon EC2 執行個體連接至 ElastiCache 叢集。在此情況下，您需要授權透過網路輸入至叢集。

### Note

如果您有使用 Local Zones，請確定已啟用。如需詳細資訊，請參閱[啟用 Local Zones](#)。如此一來，您的 VPC 會擴展到該本地區域，且您的 VPC 會將該子網路視為任何其他可用區域中的任何子網路，並自動調整相關閘道、路由表和其他安全群組考量事項。

授權從 Amazon VPC 安全群組透過網路輸入至叢集

1. 請登入 AWS Management Console，並在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，在 Network & Security (網路與安全) 下，選擇 Security Groups (安全群組)。
3. 從安全群組的清單中，選擇要用於 Amazon VPC 的安全群組。除非您建立了安全群組供 ElastiCache 使用，否則此安全群組將會命名為 default。
4. 選擇 Inbound (傳入) 標籤，然後執行下列動作：

- a. 選擇編輯。
- b. 選擇新增規則。
- c. 在 Type (類型) 欄中，選擇 Custom TCP rule (自訂 TCP 規則)。
- d. 在 Port range (連接埠範圍) 方塊中，輸入要用於叢集節點的連接埠號碼。此號碼必須與您啟動叢集時指定的號碼相同。Memcached 的預設連接埠是 **11211**。
- e. 在 Source (來源) 方塊中，選擇具有埠範圍 (0.0.0.0/0) 的 Anywhere (隨處)，讓任何您在 Amazon VPC 內啟動的 Amazon EC2 執行個體都能連接至 ElastiCache 節點。

 Important

向 0.0.0.0/0 開放 ElastiCache 叢集不會向網際網路公開該叢集，因為它沒有任何公有 IP 地址，因此無法從 VPC 外部進行存取。不過，預設安全群組可能會套用到客戶帳戶中的其他 Amazon EC2 執行個體，而這些執行個體可能有公有 IP 地址。如果他們正巧在預設連接埠上執行某些項目，就可能會意外公開該服務。因此，建議您建立一個 ElastiCache 專用的 VPC 安全群組。如需詳細資訊，請參閱[自訂安全群組](#)。

- f. 選擇 Save (儲存)。

將 Amazon EC2 執行個體啟動至 Amazon VPC 中時，該執行個體就能連接至您的 ElastiCache 叢集。

## 從 AWS 外部存取 ElastiCache 資源

Amazon ElastiCache 是一項 AWS 服務，可提供雲端型記憶體內鍵/值存放區。此服務的設計目的是為了提供 AWS 內的獨佔存取。但是，若 ElastiCache 叢集託管在 VPC 內部，您可以使用網路位址轉譯 (NAT) 執行個體來提供外部存取。

### 要求

您必須符合以下需求，才能從 AWS 外部存取 ElastiCache 資源：

- 叢集必須位於 VPC 內，並且必須透過網路位址轉譯 (NAT) 執行個體存取。此需求沒有任何例外。
- NAT 執行個體必須在與叢集相同的 VPC 內啟動。
- NAT 執行個體必須在與叢集不同的公有子網路內啟動。
- 彈性 IP 地址 (EIP) 必須與 NAT 執行個體建立關聯。iptables 的網路埠轉遞功能會用於將 NAT 執行個體上的連接埠轉送至 VPC 內的快取節點連接埠。

### 考量事項

從 ElastiCache 外部存取您的 ElastiCache 資源時，應牢記以下幾點。

- 用戶端會連線到 EIP 和 NAT 執行個體的快取連接埠。NAT 執行個體上的網路埠轉遞會將流量轉送至適當的快取叢集節點。
- 若新增或取代叢集節點，iptables 規則便需要更新以反映此變更。

### 限制

此方法應僅用於測試及開發用途。基於下列限制，不建議用於生產用途：

- NAT 執行個體的角色是用戶端與多個叢集間的代理。增加代理會影響快取叢集的效能。隨著您透過 NAT 執行個體存取的快取叢集增加，影響也會增加。
- 從用戶端到 NAT 執行個體的流量未經過加密。因此，建議您避免透過 NAT 執行個體傳送敏感性資料。
- NAT 執行個體會增加維持另一個執行個體的額外負荷。
- NAT 執行個體會做為故障的單一窗口。如需在 VPC 上設定高可用性 NAT 方式的資訊，請參閱 [Amazon VPC NAT 執行個體的高可用性：範例](#)。

## 如何從 AWS 外部存取 ElastiCache 資源

下列程序示範如何使用 NAT 執行個體連線到您的 ElastiCache 資源。

這些步驟假設下列事項：

- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6380 -j DNAT --to 10.0.1.231:6379`
- `iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 6381 -j DNAT --to 10.0.1.232:6379`

接下來，您需要反向的 NAT：

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 10.0.0.55
```

您也需要啟用 IP 轉送 (預設為停用)：

```
sudo sed -i 's/net.ipv4.ip_forward=0/net.ipv4.ip_forward=1/g' /etc/sysctl.conf sudo sysctl --system
```

- 您是透過以下設定存取 Memcached 叢集：
  - IP 地址 - 10.0.1.230
  - 預設 Memcached 連接埠 - 11211。
  - 安全群組 - \*10\0\0\55\*
- 您信任的用戶端 IP 地址為 198.51.100.27。
- 您的 NAT 執行個體具有彈性 IP 地址 203.0.113.73。
- 您的 NAT 執行個體具有安全群組 sg-ce56b7a9。

## 使用 NAT 執行個體連線到您的 ElastiCache 資源

1. 在與您快取叢集相同的 VPC，但在不同的公有子網路內建立 NAT 執行個體。

根據預設，VPC 精靈會啟動 `cache.m1.small` 節點類型。建議您根據需求選取節點大小。您必須使用 EC2 NAT AMI 才能從 AWS 外部存取 ElastiCache。

如需建立 NAT 執行個體的相關資訊，請參閱《AWS VPC 使用者指南》中的 [NAT 執行個體](#)。

2. 建立快取叢集和 NAT 執行個體的安全群組規則。

NAT 執行個體安全群組和叢集執行個體應具有以下規則：

- 兩個傳入規則
  - 其中一個允許來自信任用戶端對每個從 NAT 執行個體轉送快取連接埠的 TCP 連線 (11211 - 11213)。
  - 一個次要項目，允許對信任用戶端的 SSH 存取。

#### NAT 執行個體安全群組 - 傳入規則

Type	通訊協定	連接埠範圍	來源
自訂 TCP 規則	TCP	11211-11213	198.51.100.27/32
SSH	TCP	22	198.51.100.27/32

- 傳出規則，允許對快取連接埠 (11211) 的 TCP 連線。

#### NAT 執行個體安全群組 - 傳出規則

Type	通訊協定	連接埠範圍	目的地
自訂 TCP 規則	TCP	11211	sg-ce56b7a9 (叢集執行個體安全群組)

- 叢集安全群組的傳入規則，允許來自 NAT 執行個體對快取連接埠 (11211) 的 TCP 連線。

#### 叢集執行個體安全群組 - 傳入規則

Type	通訊協定	連接埠範圍	來源
自訂 TCP 規則	TCP	11211	sg-bd56b7da (NAT 安全群組)

### 3. 驗證規則。

- 確認信任的用戶端能夠透過 SSH 連線到 NAT 執行個體。
- 確認信任的用戶端能夠從 NAT 執行個體連線到叢集。

### 4. 將 iptable 規則新增至 NAT 執行個體。

iptables 規則必須新增到叢集中每個節點的 NAT 表，才能將來自 NAT 執行個體的快取連接埠轉送到叢集節點。範例看起來與以下內容相似：

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
10.0.1.230:11211
```

叢集中每個節點的連接埠號碼必須是唯一的。例如，若搭配連接埠 11211 到 11213 使用一個三節點 Memcached 叢集，規則看起來會與以下內容相似：

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11211 -j DNAT --to
10.0.1.230:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11212 -j DNAT --to
10.0.1.231:11211
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 11213 -j DNAT --to
10.0.1.232:11211
```

## 5. 確認信任的用戶端能夠連線到叢集。

信任的用戶端應會連線到與 NAT 執行個體相關聯的 EIP，以及對應到適當叢集節點的叢集連接埠。例如，PHP 的连接字串看起來類似如下：

```
$memcached->connect( '203.0.113.73', 11211 );
$memcached->connect( '203.0.113.73', 11212 );
$memcached->connect( '203.0.113.73', 11213 );
```

telnet 用戶端也能用於驗證連線。例如：

```
telnet 203.0.113.73 11211
telnet 203.0.113.73 11212
telnet 203.0.113.73 11213
```

## 6. 儲存 iptable 組態。

在測試及驗證過後儲存規則。如果您使用的是 Redhat 類型 Linux 發行版本 (例如 Amazon Linux)，我們建議使用以下命令：

```
service iptables save
```

## 相關主題

下列主題可能可以提供額外的資訊。

- [在 Amazon VPC 中存取 ElastiCache 快取的存取模式](#)
- [從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
- [NAT 執行個體](#)
- [設定 ElastiCache 用戶端](#)
- [Amazon VPC NAT 執行個體的高可用性：範例](#)

## 尋找連線端點

您的應用程式會使用端點連線到叢集。端點是節點或叢集的唯一位址。

### 要使用哪些端點

對於使用 Memcached 的 ElastiCache 無伺服器快取，只需從主控台取得叢集端點 DNS 和連接埠即可。

從 AWS CLI，使用 `describe-serverless-caches` 命令取得端點資訊。

### Linux

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

### Windows

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

上述操作的輸出看起來會像這樣 (JSON 格式)：

```
{
  "ServerlessCaches": [
    {
      "ServerlessCacheName": "serverless-memcached",
      "Description": "test",
      "CreateTime": 1697659642.136,
      "Status": "available",
      "Engine": "memcached",
```

```
    "MajorEngineVersion": "1.6",
    "FullEngineVersion": "21",
    "SecurityGroupIds": [
        "sg-083eda453e1e51310"
    ],
    "Endpoint": {
        "Address": "serverless-memcached-01.amazonaws.com",
        "Port": 11211
    },
    "ARN": "<the ARN>",
    "SubnetIds": [
        "subnet-0cf759df15bd4dc65",
        "subnet-09e1307e8f1560d17"
    ],
    "SnapshotRetentionLimit": 0,
    "DailySnapshotTime": "03:00"
}
]
```

**執行個體型 Memcached 叢集：**如果您使用自動探索功能，則可以使用叢集的組態端點來設定 Memcached 用戶端。這表示您必須使用支援 Automatic Discovery 的用戶端。

如果您並未使用 Automatic Discovery，則必須將用戶端設定為使用個別的節點端點以進行讀取和寫入。您也必須在新增和移除節點時追蹤這些端點。

以下各節可引導您探索執行引擎所需的端點。

## 尋找叢集的端點 (主控台)

所有 Memcached 端點都是讀取/寫入端點。若要連線至 Memcached 叢集中的節點，您的應用程式可以使用每個叢集的端點，或搭配使用 Automatic Discovery 與叢集的組態端點。若要使用 Automatic Discovery，您必須使用支援 Automatic Discovery 的用戶端。

使用 Automatic Discovery 時，您的用戶端應用程式會使用組態端點連線至 Memcached 叢集。當您透過新增或移除節點以擴展叢集時，應用程式會自動「知道」叢集中的所有節點，並能連線至其中任何節點。如果未使用 Automatic Discovery，應用程式就無法完成上述作業；您必須在每次新增或移除節點時手動更新應用程式中的端點。

若要複製端點，請選擇端點位址前面的複製圖示。如需使用端點連線至節點的相關資訊，請參閱 [連線至節點](#)。

組態端點和節點端點看起來非常類似。下列會將差異以粗體醒目提示。

```
myclustername.xxxxxx.cfg.usw2.cache.amazonaws.com:port # configuration endpoint  
contains "cfg"  
myclustername.xxxxxx.0001.usw2.cache.amazonaws.com:port # node endpoint for node 0001
```

### Important

若您選擇為 Memcached 組態端點建立正式名稱記錄 (CNAME)，為了讓自動探索用戶端將正式名稱記錄 (CNAME) 辨識為組態端點，您必須在正式名稱記錄 (CNAME) 中包含 `.cfg.`。

## 尋找端點 (AWS CLI)

您可以使用 Amazon ElastiCache 的 AWS CLI 來探索節點和叢集的端點。

### 主題

- [尋找節點和叢集的端點 \(AWS CLI\)](#)

## 尋找節點和叢集的端點 (AWS CLI)

您可以使用 AWS CLI 與 `describe-cache-clusters` 命令，來探索叢集和其節點的端點。若是 Memcached 叢集，則命令會傳回組態端點。如果您包含選用參數 `--show-cache-node-info`，則命令也會傳回叢集中個別節點的端點。

### Example

以下命令會擷取 Memcached 叢集 `mycluster` 的組態端點 (ConfigurationEndpoint) 和個別節點端點 (Endpoint)。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

上述操作的輸出看起來應會與以下內容相似 (JSON 格式)。

```
{  
  "CacheClusters": [  
    {  
      "Engine": "memcached",  
      "CacheNodes": [  
        {  
          "CacheNodeId": "0001",  
          "Endpoint": {  
            "Port": 11211,  
            "Address": "mycluster.amazonaws.com"          }  
        }  
      ]  
    }  
  ]  
}
```

```
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  },
  {
    "CacheNodeId": "0002",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  },
  {
    "CacheNodeId": "0003",
    "Endpoint": {
      "Port": 11211,
      "Address": "mycluster.amazonaws.com"
    },
    "CacheNodeStatus": "available",
    "ParameterGroupStatus": "in-sync",
    "CacheNodeCreateTime": "2016-09-22T21:30:29.967Z",
    "CustomerAvailabilityZone": "us-west-2b"
  }
],
"CacheParameterGroup": {
  "CacheNodeIdsToReboot": [],
  "CacheParameterGroupName": "default.memcached1.4",
  "ParameterApplyStatus": "in-sync"
},
"CacheClusterId": "mycluster",
"PreferredAvailabilityZone": "us-west-2b",
"ConfigurationEndpoint": {
  "Port": 11211,
  "Address": "mycluster.amazonaws.com"
},
"CacheSecurityGroups": [],
"CacheClusterCreateTime": "2016-09-22T21:30:29.967Z",
"AutoMinorVersionUpgrade": true,
"CacheClusterStatus": "available",
```

```
        "NumCacheNodes": 3,
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheSubnetGroupName": "default",
        "EngineVersion": "1.4.24",
        "PendingModifiedValues": {},
        "PreferredMaintenanceWindow": "mon:09:00-mon:10:00",
        "CacheNodeType": "cache.m4.large",
    }
]
}
```

### Important

若您選擇為 Memcached 組態端點建立 CNAME，為了讓自動探索用戶端將 CNAME 辨識為組態端點，您必須在 CNAME 中包含 `.cfg.`。例如，`session.save_path` 參數 `php.ini` 檔案中的 `mycluster.cfg.local`。

如需詳細資訊，請參閱 [describe-cache-clusters](#) 主題。

## 尋找端點 (ElastiCache API)

您可以使用 Amazon ElastiCache API 來探索節點和叢集的端點。

### 主題

- [找節點和叢集的端點 \(ElastiCache API\)](#)

### 找節點和叢集的端點 (ElastiCache API)

您可以使用 ElastiCache API 與 `DescribeCacheClusters` 動作，探索叢集和其節點的端點。若是 Memcached 叢集，則命令會傳回組態端點。如果您包含選用參數 `ShowCacheNodeInfo`，則動作也會傳回叢集中個別節點的端點。

### Example

以下命令會擷取 Memcached 叢集 `mycluster` 的組態端點 (`ConfigurationEndpoint`) 和個別節點端點 (`Endpoint`)。

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

#### Important

若您選擇為 Memcached 組態端點建立 CNAME，為了讓自動探索用戶端將 CNAME 辨識為組態端點，您必須在 CNAME 中包含 `.cfg.`。例如，`session.save_path` 參數 `php.ini` 檔案中的 `mycluster.cfg.local`。

## 管理節點

節點是 Amazon ElastiCache 部署中最小的建置區塊。它是大小固定、與網路連接的安全 RAM 區塊。每個節點都會執行當您建立或最後一次修改叢集時所選擇的引擎。每個節點都有自己的網域名稱服務

(DNS) 名稱和連接埠。支援多種類型的 ElastiCache 節點，每個節點都有各種數量的關聯記憶體和運算能力。

如需要使用何種節點大小的詳細討論，請參閱「[選擇您的 Memcached 節點大小](#)」。

## 主題

- [檢視 ElastiCache 節點狀態](#)
- [連線至節點](#)
- [支援的節點類型](#)
- [替換節點](#)
- [ElastiCache 預留節點](#)
- [遷移上一代節點](#)

與節點有關的一些重要操作如下：

- [將節點新增至叢集](#)
- [從叢集中移除節點](#)
- [縮放記憶 ElastiCache 體快取](#)
- [尋找連線端點](#)

## 檢視 ElastiCache 節點狀態

使用 [ElastiCache 控制台](#)，您可以快速訪問 ElastiCache 節點的狀態。節點的狀態表示 ElastiCache 節點的健全狀況。您可以使用下列程序在 Amazon ElastiCache 主控台、AWS CLI 命令或 API 操作中檢視 ElastiCache 節點狀態。

ElastiCache 節點的可能狀態值如下表所示。此表格也會顯示 ElastiCache 節點是否要向您收取費用。

類型	計費	Description
available	計費	ElastiCache 節點狀態良好且可用。
creating	不計費	正在建立 ElastiCache 節點。節點正在建立時無法進行存取。

類型	計費	Description
deleting	不計費	正在刪除 ElastiCache 節點。
modifying	計費	由於客戶要求修改 ElastiCache 節點，因此正在修改節點。
updating	計費	<p>「更新」狀態表示 Amazon ElastiCache 節點成立以下一或多項條件：</p> <ul style="list-style-type: none"> <li>• ElastiCache 節點正在作為服務 update 的一部分進行修補。如需服務更新的詳細資訊，請參閱 <a href="#">Amazon 受 ElastiCache 管維護和服務更新說明頁面</a>。</li> <li>• VPC 安全群組正在更新 ElastiCache 叢集。</li> <li>• ElastiCache 叢集正在 <a href="#">擴充或縮減</a>。</li> <li>• 正在修改 ElastiCache 叢集的 <a href="#">記錄傳遞組態</a>。</li> <li>• ElastiCache 節點的刪除作業正在擱置中。</li> <li>• Redis ElastiCache 的密碼正在使用更新/旋轉。 <a href="#">AWS Secrets Manager</a></li> </ul>
rebooting cache cluster nodes	計費	由於客戶請求或需要重新啟動 ElastiCache 節點的 Amazon ElastiCache 程序，節點正在重新啟動。

類型	計費	Description
incompatible_parameters	不計費	Amazon ElastiCache 無法啟動節點，因為節點參數群組中指定的參數與節點不相容。請回復參數變更，或讓參數與節點相容，以恢復節點的存取。如需有關不相容參數的詳細資訊，請檢查 ElastiCache 節點的「 <a href="#">事件</a> 」清單。
incompatible_network	不計費	不相容的網路狀態表示 Amazon 節點成立以下一或多項情況：ElastiCache <ul style="list-style-type: none"><li>• ElastiCache 節點啟動到的子網路中沒有可用的 IP 位址。</li><li>• 子網路群組中提到的 ElastiCache 子網路不再存在於 Amazon Virtual Private Cloud (Amazon VPC) 中。</li></ul>

類型	計費	Description
restore_failed	不計費	<p>恢復失敗狀態表示 Amazon 節點成立以下情況之一：</p> <p>ElastiCache</p> <ul style="list-style-type: none"> <li>由於重複發生<u>執行個體容量不足</u>的情況，因此更換節點失敗。這通常會在執行上一代節點時發生 end-of-life。但是，AWS 如果沒有足夠的隨需容量在指定的可用區域中滿足您的請求，則更換最新一代節點也可能發生這種情況。如需修正或移除這些節點的詳細資訊，請參閱<u>遷移上一代節點</u>。</li> <li>指定的 RDB 快照還原失敗。</li> <li>ElastiCache 叢集的 AWS 帳戶已暫停。</li> <li>節點發生故障，無法復原。</li> </ul>
snapshotting	計費	ElastiCache 正在建立 Redis 節點 ElastiCache 的快照集。

## 使用主控台檢視 ElastiCache 節點狀態

要使用控制台查看 ElastiCache 節點的狀態：

- 登錄到 AWS Management Console 並打開 Amazon ElastiCache 控制台 <https://console.aws.amazon.com/elasticache/>。
- 在瀏覽窗格中，選擇 Redis 叢集或記憶體快取叢集。便會顯示「快取記憶體」頁面，內含 ElastiCache 節點清單。每個節點的狀態值都會顯示。
- 然後，您可以瀏覽至快取的 [服務更新] 索引標籤，以顯示適用於快取的服務更新清單。

## 檢視 ElastiCache 節點狀態 AWS CLI

若要使用檢視 ElastiCache 節點及其狀態資訊 AWS CLI，請使用 `describe-cache-cluster` 指令。例如，下列 AWS CLI 指令會顯示每個 ElastiCache 節點。

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

## 透過 API 檢視 ElastiCache 節點狀態

若要使用 Amazon ElastiCache API 檢視 ElastiCache 節點的狀態，請使用 `ShowCacheNodeInfo` 旗標呼叫以擷取 `DescribeCacheClusteroperation` 有關個別快取節點的資訊。

## 連線至節點

在嘗試連線到您的 Memcached 叢集前，您必須擁有節點的端點。若要尋找端點，請參閱下列主題：

- [尋找叢集的端點 \(主控台\)](#)
- [尋找端點 \(AWS CLI\)](#)
- [尋找端點 \(ElastiCache API\)](#)

在下列範例中，您會使用 telnet 公用程式連線到執行 Memcached 的節點。

### Note

如需 Memcached 和可用 Memcached 命令的詳細資訊，請參閱 [Memcached](#) 網站。

### 使用 telnet 連線到節點

1. 使用您選擇的連線公用程式連線到 Amazon EC2 執行個體。

### Note

如需如何連線到 Amazon EC2 執行個體的說明，請參閱 [Amazon EC2 入門指南](#)。

2. 下載 telnet 公用程式並安裝在 Amazon EC2 執行個體上。在 Amazon EC2 執行個體的命令提示中輸入下列命令，然後在命令提示中輸入 y。

```
sudo yum install telnet
```

隨即出現類似如下的輸出。

```
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. 在 Amazon EC2 執行個體的命令提示中輸入下列命令，並以您的節點端點取代此範例中所示的內容。

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

隨即出現類似如下的輸出。

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

4. 透過執行 Memcached 命令來測試連線。

您現在已連線至節點，可以執行 Memcached 命令。以下是範例。

```
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
>
```

## 支援的節點類型

如需要使用何種節點大小的詳細資訊，請參閱「[選擇您的 Memcached 節點大小](#)」。

ElastiCache 支持以下節點類型。一般而言，最新一代的節點類型跟前一代相同的節點類型比較起來，能夠以較低的成本提供更多記憶體和運算能力。

如需深入了解每個節點類型的效能詳細資訊，請參閱 [Amazon EC2 執行個體類型](#)。

- 一般用途：
- 最新一代：

M6g 節點類型 (僅適用於 Memcached 引擎 1.5.16 以上版本)。

cache.m6g.large, cache.m6g.xlarge, cache.m6g.2xlarge, cache.m6g.4xlarge, cache.m6g.8xlarge, cache.m6g.12xlarge, cache.m6g.16xlarge

### Note

如需區域可用性，請參閱「[支援的節點類型 \(依 AWS 區域\)](#)」。

### M5 節點類

型：cache.m5.large、cache.m5.xlarge、cache.m5.2xlarge、cache.m5.4xlarge、cache.

### M4 節點類

型：cache.m4.large、cache.m4.xlarge、cache.m4.2xlarge、cache.m4.4xlarge、cache.

T4g 節點類型 (僅適用於 Memcached 引擎 1.5.16 以上版本)。

cache.t4g.micro, cache.t4g.small, cache.t4g.medium

T3 節點類型：cache.t3.micro、cache.t3.small、cache.t3.medium

T2 節點類型：cache.t2.micro、cache.t2.small、cache.t2.medium

- 前一代：(不建議。仍然支援現有叢集，但這些類型不支援建立新叢集。)

T1 節點類型：cache.t1.micro

### M1 節點類

型：cache.m1.small、cache.m1.medium、cache.m1.large、cache.m1.xlarge

### M3 節點類

型：`cache.m3.medium`、`cache.m3.large`、`cache.m3.xlarge`、`cache.m3.2xlarge`

- 運算優化:

- 前一代：(不建議)

C1 節點類型：`cache.c1.xlarge`

- 記憶體優化:

- 最新一代：

(R6g 節點類型僅適用於 Memcached 引擎 1.5.16 以上版本。)

### R6g 節點類

型：`cache.r6g.large`、`cache.r6g.xlarge`、`cache.r6g.2xlarge`、`cache.r6g.4xlarge`、`cache.r6g.8xlarge`

#### Note

如需區域可用性，請參閱「[支援的節點類型 \(依 AWS 區域\)](#)」。

### R5 節點類

型：`cache.r5.large`、`cache.r5.xlarge`、`cache.r5.2xlarge`、`cache.r5.4xlarge`、`cache.r5.8xlarge`

### R4 節點類

型：`cache.r4.large`、`cache.r4.xlarge`、`cache.r4.2xlarge`、`cache.r4.4xlarge`、`cache.r4.8xlarge`

- 前一代：(不建議)

M2 節點類型：`cache.m2.xlarge`、`cache.m2.2xlarge`、`cache.m2.4xlarge`

### R3 節點類

型：`cache.r3.large`、`cache.r3.xlarge`、`cache.r3.2xlarge`、`cache.r3.4xlarge`、`cache.r3.8xlarge`

- 網路最佳化：

- 最新一代：

(C7gn 節點類型僅適用於 Memcached 引擎 1.6.6 以上版本。)

### C7gn 節點類

型：`cache.c7gn.large`、`cache.c7gn.xlarge`、`cache.c7gn.2xlarge`、`cache.c7gn.4xlarge`、`cache.c7gn.8xlarge`

## 最新一代

下列表單顯示使用網路 I/O 抵用金機制來超越其基準頻寬之執行個體類型的基準頻寬和高載頻寬。

### Note

具有爆量網路效能的執行個體類型使用網路 I/O 點數機制，以竭盡全力爆量到其基準頻寬之外。

### 一般

執行個體類型	支援的最低 Memcached 版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m7g.large		0.937	12.5
cache.m7g.xlarge		1.876	12.5
cache.m7g.2xlarge		3.75	15
cache.m7g.4xlarge		7.5	15
cache.m7g.8xlarge		15	N/A
cache.m7g.12xlarge		22.5	N/A
cache.m7g.16xlarge		30	N/A
cache.m6g.large	1.5.16	0.75	10.0
cache.m6g.xlarge	1.5.16	1.25	10.0
cache.m6g.2xlarge	1.5.16	2.5	10.0
cache.m6g.4xlarge	1.5.16	5.0	10.0
cache.m6g.8xlarge	1.5.16	12	N/A
cache.m6g.12xlarge	1.5.16	20	N/A

執行個體類型	支援的最低 Memcached 版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.m6g.16xlarge	1.5.16	25	N/A
cache.m5.large	1.5.16	0.75	10.0
cache.m5.xlarge	1.5.16	1.25	10.0
cache.m5.2xlarge	1.5.16	2.5	10.0
cache.m5.4xlarge	1.5.16	5.0	10.0
cache.m5.12xlarge	1.5.16	N/A	N/A
cache.m5.24xlarge	1.5.16	N/A	N/A
cache.m4.large	1.5.16	0.45	1.2
cache.m4.xlarge	1.5.16	0.75	2.8
cache.m4.2xlarge	1.5.16	1.0	10.0
cache.m4.4xlarge	1.5.16	2.0	10.0
cache.m4.10xlarge	1.5.16	5.0	10.0
cache.t4g.micro	1.5.16	0.064	5.0
cache.t4g.small	1.5.16	0.128	5.0
cache.t4g.medium	1.5.16	0.256	5.0
cache.t3.micro	1.5.16	0.064	5.0
cache.t3.small	1.5.16	0.128	5.0
cache.t3.medium	1.5.16	0.256	5.0
cache.t2.micro	1.5.16	0.064	1.024
cache.t2.small	1.5.16	0.128	1.024

執行個體類型	支援的最低 Memcached 版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.t2.medium	1.5.16	0.256	1.024

## 記憶體最佳化

執行個體類型	最低支援版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r7g.large		0.937	12.5
cache.r7g.xlarge		1.876	12.5
cache.r7g.2xlarge		3.75	15
cache.r7g.4xlarge		7.5	15
cache.r7g.8xlarge		15	N/A
cache.r7g.12xlarge		22.5	N/A
cache.r7g.16xlarge		30	N/A
cache.r6g.large	1.5.16	0.75	10.0
cache.r6g.xlarge	1.5.16	1.25	10.0
cache.r6g.2xlarge	1.5.16	2.5	10.0
cache.r6g.4xlarge	1.5.16	5.0	10.0
cache.r6g.8xlarge	1.5.16	12	N/A
cache.r6g.12xlarge	1.5.16	20	N/A
cache.r6g.16xlarge	1.5.16	25	N/A
cache.r5.large	1.5.16	0.75	10.0
cache.r5.xlarge	1.5.16	1.25	10.0

執行個體類型	最低支援版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.r5.2xlarge	1.5.16	2.5	10.0
cache.r5.4xlarge	1.5.16	5.0	10.0
cache.r5.12xlarge	1.5.16	20	N/A
cache.r5.24xlarge	1.5.16	25	N/A
cache.r4.large	1.5.16	0.75	10.0
cache.r4.xlarge	1.5.16	1.25	10.0
cache.r4.2xlarge	1.5.16	2.5	10.0
cache.r4.4xlarge	1.5.16	5.0	10.0
cache.r4.8xlarge	1.5.16	12	N/A
cache.r4.16xlarge	1.5.16	25	N/A

## 網路最佳化

執行個體類型	最低支援版本	基準頻寬 (Gbps)	高載頻寬 (Gbps)
cache.c7gn.large	1.6.6	6.25	30
cache.c7gn.xlarge	1.6.6	12.5	40
cache.c7gn.2xlarge	1.6.6	25	50
cache.c7gn.4xlarge	1.6.6	50	N/A
cache.c7gn.8xlarge	1.6.6	100	N/A
cache.c7gn.12xlarge	1.6.6	150	N/A
cache.c7gn.16xlarge	1.6.6	200	N/A

## 支援的節點類型 (依 AWS 區域)

支援的節點類型可能因區 AWS 域而異。如需詳細資訊，請參閱 [Amazon ElastiCache 定價](#)。

### 爆量效能執行個體

您可以在 Amazon 上啟動通用高載 T4G，T3 標準和 T2 標準緩存節點。ElastiCache 這些節點提供 CPU 基準效能，能夠讓 CPU 使用率隨時火力全開，直至累積的點數耗盡。一個 CPU 點數提供一分鐘、一個 CPU 核心的完整效能。

Amazon ElastiCache 的 T4G、T3 和 T2 節點設定為標準節點，適用於 CPU 平均使用率持續低於執行個體基準效能的工作負載。若要讓效能突增至超過基準，節點會用掉在 CPU 點數餘額中所累積的點數。若節點的累積點數不足，效能會逐漸降低至基準效能等級。逐漸降低效能可確保節點的累積 CPU 點數餘額耗盡時，效能不會急速下滑。如需詳細資訊，請參閱 Amazon EC2 Linux 執行個體使用者指南中的 [爆量效能執行個體的 CPU 點數與基準效能](#)。

下表列出高載效能節點類型，以及 CPU 點數的每小時獲得速率。其中也顯示節點可累積獲得的 CPU 點數最大值，以及每個節點的 vCPU 數。此外也提供相較於完整核心效能 (使用單一 vCPU)，基準效能等級的百分比。

每小時獲得的 CPU 額度	可累積的點數上限*	vCPU	每個 vCPU 的基準效能	記憶體 (GiB)	網路效能
12	288	2	10%	0.5	最多 5 GB
24	576	2	20%	1.37	最多 5 GB
24	576	2	20%	3.09	最多 5 GB
12	288	2	10%	0.5	最多 5 GB
24	576	2	20%	1.37	最多 5 GB
24	576	2	20%	3.09	最多 5 GB
6	144	1	10%	0.5	低至中度

每小時獲得的 CPU 額度	可累積的點數上限*	vCPU	每個 vCPU 的基準效能	記憶體 (GiB)	網路效能
12	288	1	20%	1.55	低至中度
24	576	2	20%	3.22	低至中度

\* 可累計的額度等於在 24 小時期間可獲得的額度。

\*\* 表格中所顯示的是每 vCPU 的基準效能。部分節點大小包含不只一個 vCPU。若要計算這些節點的基準 CPU 使用率，請將 vCPU 的百分比乘上 vCPU 的數量。

T3 高載效能執行個體提供下列 CPU 點數指標：

 Note

這些指標不適用於 T2 高載效能執行個體。

- CPUCreditUsage
- CPUCreditBalance

如需這些指標的詳細資訊，請參閱 [CPU 點數指標](#)。

此外，也請注意這些細節：

- 所有目前一代的節點類型預設都建立在以 Amazon VPC 為基礎的 Virtual Private Cloud (VPC)。

## 相關資訊

- [Amazon ElastiCache 產品功能和細節](#)
- [Memcached 節點類型限定參數](#)

## 替換節點

Amazon ElastiCache for Memcached 會透過將修補程式和升級順暢套用到執行個體，來頻繁升級其機群。不過，我們不時需要重新啟動您的 ElastiCache for Memcached 節點，以便對基礎主機套用強制 OS 更新。必須進行這些替換才能套用升級，以強化安全、可靠性和操作效能。

您可以選擇在排程的節點替換時間之前，隨時自行管理這些替換。當您自行管理替換時，執行個體會重新啟動節點時收到 OS 更新，而排程的節點替換將會取消。您可能會繼續收到提醒，指出節點即將進行替換。若您已手動減少維護的需求，您可以忽略這些提醒。

### Note

Amazon ElastiCache 自動產生的取代快取節點可能有不同的 IP 地址。您必須負責檢閱應用程式組態，以確保快取節點與適當的 IP 位址相關聯。

以下清單會指出在 ElastiCache 排程替換掉其中一個 Memcached 節點時，您可以採取的動作。

- 不執行任何動作 - 若您不執行任何動作，ElastiCache 會依排程替換節點。當 ElastiCache 自動使用新節點替換節點時，新的節點一開始會處於空白狀態。
- 變更您的維護時段 - 對於排定的維護事件，您會收到來自 ElastiCache 的電子郵件或通知事件。在這種情況下，如果在排定的替換時間之前變更維護時段，則現在將在新的時間替換您的節點。如需更多詳細資訊，請參閱 [修改 ElastiCache 叢集](#)。

### Note

只有當 ElastiCache 通知包含維護時段時，您才可以透過改變維護時段來變更您的替換時間。若通知並未包含維護時段，您便無法變更替換時間。

例如，假設現在是 11 月 9 日星期四下午 3:00，下一個維護時段是 11 月 10 日星期五下午 5:00。以下是三種情況及其結果：

- 您將維護時段變更為星期五下午 4:00，在目前的日期時間之後、下一個排定的維護時段之前。節點將於 11 月 10 日星期五下午 4:00 進行替換。
- 您將維護時段變更為星期六下午 4:00，在目前的日期時間之後，以及下一個排定的維護時段之後。節點將於 11 月 11 日星期六下午 4:00 進行替換。
- 您將維護時段變更為星期三下午 4:00，早於這星期中目前的日期時間。節點將於 11 月 15 日星期三下午 4:00 進行替換。

如需說明，請參閱 [管理維護作業](#)。

- 手動替換節點 - 若您需要在下一個維護時段之前替換節點，請手動替換節點。

如果手動替換節點，則會重新配送金鑰。此重新配送會導致快取遺漏。

#### 手動替換 Memcached 節點

1. 刪除已排程替換的節點。如需說明，請參閱 [從叢集中移除節點](#)。
2. 將新的節點新增到叢集。如需說明，請參閱 [將節點新增至叢集](#)。
3. 若您並未在此叢集上使用自動探索，請查看您的應用程式並使用新節點的端點替換舊節點端點的每個執行個體。

## ElastiCache 預留節點

預留一或多個節點可能可以減少您的成本。您必須先為預留節點預付費用，實際費用取決於節點類型及保留時間長度 (一或三年)。

若要了解預留節點在您的使用案例中是否可以節省成本，請先判斷您需要的節點大小和節點數量。然後估計節點的用量，並比較使用隨需節點和預留節點時的總成本。您可以在您的叢集中混合搭配使用預留節點與隨需節點。如需定價資訊，請參閱 [Amazon ElastiCache 定價](#)。

### Note

預留節點並不靈活；它們只適用於您保留的確切執行個體類型。

### 管理使用預留節點的成本

預留一或多個節點可能可以減少您的成本。您必須先為預留節點預付費用，實際費用取決於節點類型及保留時間長度 (一或三年)。此費用遠低於您使用隨需節點時需支付的每小時使用費。

若要了解預留節點在您的使用案例中是否可以節省成本，請先判斷您需要的節點大小和節點數量。然後估計節點的用量，並比較使用隨需節點和預留節點時的總成本。您可以在您的叢集中混合搭配使用預留節點與隨需節點。如需定價資訊，請參閱 [Amazon ElastiCache 定價](#)。

購買時必須選擇 AWS 區域、節點類別和期限長度，這些日後無法變更。

您可以使用 AWS Management Console、AWS CLI 或 ElastiCache API 來列出和購買可用的預留節點方案。

如需預留節點的詳細資訊，請參閱 [Amazon ElastiCache 預留節點](#)。

### 主題

- [標準預留節點方案](#)
- [舊式預留節點方案](#)
- [取得關於預留節點方案的資訊](#)
- [購買預留節點](#)
- [取得預留節點的相關資訊](#)

## 標準預留節點方案

在 Amazon ElastiCache 中購買標準預留節點執行個體 (RI) 時，您必須綁約，以在預留節點執行個體有效期間享有特定節點執行個體類型和 AWS 區域的折扣費率。若要使用 Amazon ElastiCache 預留節點執行個體，您應建立新的 ElastiCache 節點執行個體，如同建立隨需執行個體一般。

您建立的新節點執行個體必須完全符合預留節點執行個體的規格。如果新節點執行個體規格與您帳戶現有的預留節點執行個體相符，您將以預留節點執行個體的折扣費率計費。否則，節點執行個體將按隨需費率計費。這些標準 RI 可供 R5 和 M5 執行個體系列以上使用。

### Note

接下來討論的這三個方案類型皆提供一年期和三年期。

## 方案類型

無須預付 RI 讓您不用支付預付款，便能存取預留 ElastiCache 執行個體。無須預付的預留 ElastiCache 執行個體在期間中以折扣後的每小時費率計費，無論是否有使用。

部分預付 RI 要求預先支付部分的預留 ElastiCache 資料庫執行個體。期間內其餘的時數會以折扣後的每小時費率計費，無論是否有使用。此選項為傳統重度使用選項的替代方案，下一節將說明。

全部預付 RI 要求在 RI 有效期間開始時繳付全額款項。有效期間開始時便支付全額款項，且有效期間的剩餘時間內，無論使用多少小時，都不會產生其他成本或額外的每小時費用。

## 舊式預留節點方案

有三個層級的舊式節點保留：重度使用率、中度使用率和輕度使用率。節點可以在任何使用率層級保留 1 或 3 年。節點類型、使用率層級和保留有效期間會影響您的總成本。購買預留節點之前，請比較各種模型，確認預留節點可為您的業務省下的費用。

在某個使用率層級或期間購買的節點不能轉換為不同的使用率層級或期間。

## 使用率層級

重度使用率預留節點允許具有一致容量基線的工作負載或執行穩定狀態的工作負載。重度使用率預留節點需要高度的前期承諾，但如果您計劃執行多於 79% 的預留節點期間，您可以獲得最高的節省金額 (節省金額最高達隨需價格的 70%)。若採用高度使用率預留節點，您需支付一筆一次性費用。之後無論您的節點是否有執行，再有效期間內都需支付較低的每小時費用。

如果您計劃運用預留節點很長一段時間，且您需要較低的一次性費用，或希望可在節點關閉時停止支付其費用，則中度使用率預留節點為最佳選項。當您計劃執行超過 40% 的預留節點期間，則中度使用率預留節點為更符合成本效益的選項。此選項可以為您節省高達隨需價格的 64%。利用中度使用率預留節點，您需支付較輕度使用率預留節點稍高的一次性費用，並且會在執行節點時獲得較低的每小時使用費率。

輕度使用率預留節點為一天只會執行幾個小時，或一週執行幾天的定期性工作負載的理想選擇。使用輕度使用率預留節點，您會支付一次性費用，接著是您的節點執行時每小時折扣的使用費。當您的節點執行時間超過預留節點有效期間的 17%，您就可以開始節省成本。預留節點的整個有效期間內，您可以節省高達隨需費率的 56%。

### 舊式預留節點方案

產品	預付成本	使用費	優點
高度使用率	最高	每小時費用最低。無論您是否使用預留節點，會套用至整個期間。	如果您計劃執行預留節點超過三年期間的 79%，則整體成本最低。
中度使用率	中型	對執行節點的每個小時收取每小時使用費。節點未執行時不收取每小時費用。	適合彈性工作負載或當您預期的是中度使用量 (超過三年期間的 40%)。
低度使用率	最低	對執行節點的每個小時收取每小時使用費。節點未執行時不收取每小時費用。所有方案類型的每小時費用最高，但費用僅套用在預留節點執行時。	如果您規劃不間斷執行，則整體成本會最高。不過，如果您規劃的預留節點使用頻率較低 (三年有效期間大約超過 15%)，則整體成本最低。
隨需使用 (無預留節點)	無		每小時成本最高。

產品	預付成本	使用費	優點
		每小時費用最高。每當節點執行時便套用。	

如需詳細資訊，請參閱 [Amazon ElastiCache 定價](#)。

## 取得關於預留節點方案的資訊

購買預留節點之前，您可以取得關於可用的預留節點方案的資訊。

下列範例顯示如何使用 AWS Management Console、AWS CLI 和 ElastiCache API 來取得可用預留節點方案的定價和相關資訊。

### 主題

- [取得關於預留節點方案的資訊 \(主控台\)](#)
- [取得關於預留節點方案的資訊 \(AWS CLI\)](#)
- [取得關於預留節點方案的資訊 \(ElastiCache API\)](#)

### 取得關於預留節點方案的資訊 (主控台)

若要使用 AWS Management Console 來取得可用預留叢集方案的定價和其他資訊，請使用下列程序。

#### 取得可用預留節點方案的相關資訊

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中，選擇 Reserved Instances (預留節點)。
3. 選擇 Purchase Reserved Nodes (購買預留節點)。
4. 針對 Engine (引擎)，選擇 Memcached。
5. 若要判斷可用的方案，請為接下來的選項進行選擇：
  - 節點類型
  - 期間
  - 方案類型

進行這些選擇之後，Reservation details (預留詳細資訊) 中會顯示每一節點的成本和您選擇的總成本。

6. 選擇 Cancel (取消) 來避免購買這些節點和衍生費用。

### 取得關於預留節點方案的資訊 (AWS CLI)

若要取得可用預留節點方案的定價和其他資訊，在命令提示字元中，輸入下列命令：

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

此作業會產生類似下列的輸出 (JSON 格式)：

```
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.large",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "memcached",
  "OfferingType": "All Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.X,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xxx.xlarge",
  "Duration": 94608000,
  "FixedPrice": XXXX.X,
  "UsagePrice": X.X,
  "ProductDescription": "memcached",
  "OfferingType": "Partial Upfront",
  "RecurringCharges": [
    {
      "RecurringChargeAmount": X.XXXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
},
{
  "ReservedCacheNodesOfferingId": "0xxxxxxxx-xxeb-44ex-xx3c-xxxxxxxx072",
  "CacheNodeType": "cache.xx.12xlarge",
  "Duration": 31536000,
  "FixedPrice": X.X,
  "UsagePrice": X.X,
  "ProductDescription": "memcached",
  "OfferingType": "No Upfront",
  "RecurringCharges": [
```

```
    {
      "RecurringChargeAmount": X.XXXX,
      "RecurringChargeFrequency": "Hourly"
    }
  ]
}
```

如需詳細資訊，請參閱「AWS CLI 參考」中的 [describe-reserved-cache-nodes-offerings](#)。

取得關於預留節點方案的資訊 (ElastiCache API)

若要取得關於可用的預留節點方案的定價和資訊，請呼叫 `DescribeReservedCacheNodesOfferings` 動作。

### 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>
```

如需詳細資訊，請參閱「ElastiCache API 參考」中的 [DescribeReservedCacheNodesOfferings](#)。

## 購買預留節點

下列範例顯示如何使用 AWS Management Console、AWS CLI 和 ElastiCache API 來購買預留節點方案。

### Important

若依照本節中的範例操作，將對您的 AWS 帳戶衍生費用且無法還原。

## 主題

- [購買預留節點 \(主控台\)](#)
- [購買預留節點 \(AWS CLI\)](#)
- [購買預留節點 \(ElastiCache API\)](#)

## 購買預留節點 (主控台)

此範例顯示購買特定的預留節點方案 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f，並且其預留節點 ID 為 myreservationID。

下列程序使用 AWS Management Console 來依方案識別符購買預留節點方案。

## 購買預留節點

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽清單中，選擇 Reserved Nodes (預留節點) 連結。
3. 選擇 Purchase Reserved Node (購買預留節點) 按鈕。
4. 針對 Engine (引擎)，選擇 Memcached。
5. 若要判斷可用的方案，請為接下來的選項進行選擇：
  - 節點類型
  - 期間
  - 方案類型
  - 選擇性的 Reserved node ID (預留節點識別碼)

進行這些選擇之後，Reservation details (預留詳細資訊) 中會顯示每一節點的成本和您選擇的總成本。

## 6. 選擇 Purchase (購買)。

### 購買預留節點 (AWS CLI)

下列範例顯示購買特定的預留叢集方案 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f，並且其預留節點 ID 為 myreservationID。

在命令提示字元中，輸入下列命令：

若為 Linux、macOS 或 Unix：

```
aws elasticache purchase-reserved-cache-nodes-offering \  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \  
  --reserved-cache-node-id myreservationID
```

針對 Windows：

```
aws elasticache purchase-reserved-cache-nodes-offering ^  
  --reserved-cache-nodes-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^  
  --reserved-cache-node-id myreservationID
```

此命令會傳回類似以下的輸出：

RESERVATION	ReservationId	Class	Start Time	Duration	
Fixed Price	Usage Price	Count	State	Description	Offering Type
RESERVATION	myreservationid	cache.xx.small	2013-12-19T00:30:23.247Z	1y	
XXX.XX USD	X.XXX USD	1	payment-pending	memcached	Medium Utilization

如需詳細資訊，請參閱「AWS CLI 參考」中的 [purchase-reserved-cache-nodes-offering](#)。

### 購買預留節點 (ElastiCache API)

下列範例顯示購買特定的預留節點方案 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f，並且其預留叢集 ID 為 myreservationID。

搭配下列參數呼叫 PurchaseReservedCacheNodesOffering 操作：

- ReservedCacheNodesOfferingId = 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f
- ReservedCacheNodeID = myreservationID
- CacheNodeCount = 1

## Example

```
https://elasticache.us-west-2.amazonaws.com/  
  ?Action=PurchaseReservedCacheNodesOffering  
  &ReservedCacheNodesOfferingId=649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  
  &ReservedCacheNodeID=myreservationID  
  &CacheNodeCount=1  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z  
  &X-Amz-Algorithm=&AWS;4-HMAC-SHA256  
  &X-Amz-Date=20141201T220302Z  
  &X-Amz-SignedHeaders=Host  
  &X-Amz-Expires=20141201T220302Z  
  &X-Amz-Credential=<credential>  
  &X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱「ElastiCache API 參考」中的 [PurchaseReservedCacheNodesOffering](#)。

## 取得預留節點的相關資訊

您可以使用 AWS Management Console、AWS CLI 和 ElastiCache API，取得您已購買之預留節點的相關資訊。

### 主題

- [取得預留節點的相關資訊 \(主控台\)](#)
- [取得預留節點的相關資訊 \(AWS CLI\)](#)
- [取得預留節點的相關資訊 \(ElastiCache API\)](#)

### 取得預留節點的相關資訊 (主控台)

下列程序描述如何使用 AWS Management Console 來取得您購買之預留節點的相關資訊。

#### 取得您購買之預留節點的相關資訊

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽清單中，選擇 Reserved Nodes (預留節點) 連結。

帳戶的預留節點會出現在預留節點清單中。選擇清單中的任何預留節點，便能從主控台底部的詳細資料窗格中查看該預留節點的詳細資訊。

### 取得預留節點的相關資訊 (AWS CLI)

若要取得 AWS 帳戶預留節點的相關資訊，請在命令提示字元中輸入下列命令：

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

此操作會產生類似下列的輸出 (JSON 格式)：

```
{
  "ReservedCacheNodeId": "myreservationid",
  "ReservedCacheNodesOfferingId": "649fd0c8-cf6d-47a0-bfa6-060f8e75e95f",
  "CacheNodeType": "cache.xx.small",

  "Duration": "31536000",
  "ProductDescription": "memcached",
  "OfferingType": "Medium Utilization",
```

```
"MaxRecords": 0
}
```

如需詳細資訊，請參閱「AWS CLI 參考」中的 [describe--reserved-cache-nodes](#)。

取得預留節點的相關資訊 (ElastiCache API)

若要取得 AWS 帳戶之預留節點的相關資訊，請呼叫 DescribeReservedCacheNodes 作業。

### Example

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

如需詳細資訊，請參閱「ElastiCache API 參考」中的 [DescribeReservedCacheNodes](#)。

## 遷移上一代節點

上一代節點是逐步遭淘汰的節點類型。如果您現有的叢集沒有使用上一代節點類型，ElastiCache 不支援建立具有此節點類型的新叢集。

由於上一代節點類型的數量有限，我們無法保證您叢集中的節點運作狀態不良時可成功替換。在這種情況下，您的叢集可用性可能會受到負面影響。

建議您將叢集遷移至新的節點類型，以獲得更佳的可用性和效能。如需建議的遷移節點類型，請參閱[升級途徑](#)。如需 ElastiCache 支援的節點類型和上一代節點類型的完整清單，請參閱「[支援的節點類型](#)」。

## 遷移 Memcached 叢集上的節點

若要將 ElastiCache for Memcached 遷移到不同的節點類型，您必須建立一個新的叢集，這類叢集一開始一律是空的，以便填入您的應用程式。

## 使用 ElastiCache 主控台遷移 ElastiCache for Memcached 叢集節點類型：

- 使用新的節點類型來建立新叢集。如需更多詳細資訊，請參閱 [建立 Memcached 叢集 \(主控台\)](#)。
- 在應用程式中，將端點更新為新叢集的端點。如需詳細資訊，請參閱「[尋找叢集的端點 \(主控台\)](#)」
- 刪除舊叢集。如需詳細資訊，請參閱 [刪除叢集](#)

# 使用 ElastiCache

在本節中，您可以找到有關如何管理 ElastiCache 實施的各種組件的詳細信息。

## 主題

- [快照和還原](#)
- [引擎版本和升級](#)
- [ElastiCache 最佳實踐和緩存策略](#)
- [管理您的自行設計叢集](#)
- [縮放記憶 ElastiCache 體快取](#)
- [為您的 ElastiCache 資源加上標籤](#)
- [使用 Amazon ElastiCache Well-Architected Lens](#)
- [常見疑難排解步驟和最佳做法](#)
- [其他疑難排解步](#)

## 快照和還原

執行 無伺服器 Memcached 的 Amazon ElastiCache 快取可透過建立快照來備份其資料。您可以使用備份來還原快取或將資料植入新的快取。備份包含快取的中繼資料，以及快取中的所有資料。所有備份都會寫入 Amazon Simple Storage Service (Amazon S3)，該服務提供耐久性儲存空間。您可以隨時建立新的 無伺服器 Memcached 快取，並將備份中的資料填入資料來還原資料。使用 ElastiCache，您可以使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 和 ElastiCache API 來管理備份。

如果您打算刪除快取，而且有必要保留資料，您可以採取額外的預防措施。若要達成此目標，請先建立手動備份、確認其狀態為可用，然後刪除快取。這樣做可確保備份失敗時，快取資料仍然可用。您可以依照前述的最佳實務，重試進行備份。

## 主題

- [備份限制](#)
- [排程自動備份](#)
- [取得手動備份](#)
- [建立最終備份](#)
- [描述備份](#)

- [複製備份](#)
- [從備份還原到新的快取](#)
- [刪除備份](#)
- [標記備份](#)

## 備份限制

規劃或建立備份時，請考慮下列限制：

- 只有在 Redis 或無伺服器 Memcached 上執行的快取才支援 Backup 和還原。
- 在任何連續的 24 小時期間內，每個無伺服器快取最多可建立 24 個手動備份。
- 在備份過程中，您無法在無伺服器快取上執行任何其他 API 或 CLI 作業。

## 排程自動備份

您可以為任何 Memcached 無伺服器快取啟用自動備份。啟用自動備份後，ElastiCache 會每天建立快取備份。不會對快取造成任何影響，而且變更會立即生效。自動備份可協助防止資料遺失。如果發生失敗，您可以建立新的快取，從最近的備份還原您的資料。這會產生暖啟動的快取，並預先載入您的資料以供使用。如需詳細資訊，請參閱 [從備份還原到新的快取](#)。

當您排程自動備份時，您 ElastiCache 將何時開始建立備份。您可以將備份時段設定在最方便的任何時間。如果您未指定備份時段，請自動 ElastiCache 指定一個備份時段。

您可以在建立 Memcached 無伺服器快取時啟用或停用自動備份，方法是使用 ElastiCache 主控台、AWS CLI、或 API。ElastiCache 這是通過檢查啟用自動備份在高級 Memcached 設置部分中的框來完成。

## 取得手動備份

除了自動備份，您還可以隨時建立「手動」備份。不同於自動備份會在指定的保留期之後自動刪除，手動備份沒有過了一段時間會自動刪除的保留期。即使您刪除快取，該快取中的任何手動備份仍會保留。如果您不想再保留手動備份，則必須自行明確予以刪除。

除了直接建立手動備份，您還可以利用下列其中一種方式來建立手動備份：

- [複製備份](#)。與來源備份是自動或手動建立無關。
- [建立最終備份](#)。在刪除叢集或節點前立即建立一個備份。

您可以使用 AWS Management Console、或 ElastiCache API 建立手動備份快取。AWS CLI

### 建立手動備份 (主控台)

### 建立快取的備份 (主控台)

1. 登入 AWS Management Console 並開啟 Amazon EC2 主控台，網址為 <https://console.aws.amazon.com/ec2/>。
2. 在導航窗格中，選擇內存緩存。
3. 選擇您要備份的快取名稱左側的方塊。
4. 選擇 Backup (備份)。
5. 在 Create Backup (建立備份) 對話方塊的 Backup Name (備份名稱) 方塊中，輸入您的備份名稱。我們建議名稱指出備份的叢集，以及建立備份的日期和時間。

叢集命名限制條件如下：

- 必須包含 1-40 個英數字元或連字號。
  - 必須以字母開頭。
  - 不能連續包含兩個連字號。
  - 結尾不能是連字號。
6. 選擇 Create Backup (建立備份)。

叢集的狀態會變更為「快照中」。

## 建立手動備份 (AWS CLI)

### 使用手動備份無伺服器快取 AWS CLI

若要使用建立快取的手動備份 AWS CLI，請使用具有下列參數的 `create-serverless-snapshot` AWS CLI 作業：

- `--serverless-cache-name` - 您要備份之無伺服器快取的名稱。
- `--serverless-cache-snapshot-name` - 要建立的快照名稱。

若為 Linux、macOS 或 Unix：

- ```
aws elasticache create-serverless-snapshot \  
    --serverless-cache-name CacheName \  
    --serverless-cache-snapshot-name bkup-20231127
```

針對 Windows：

- ```
aws elasticache create-serverless-snapshot ^  
    --serverless-cache-name CacheName ^  
    --serverless-cache-snapshot-name bkup-20231127
```

## 相關主題

如需詳細資訊，請參閱 AWS CLI 命令參考中的 [create-snapshot](#)。

## 建立最終備份

您可以使用 ElastiCache 主控台、AWS CLI、或 ElastiCache API 建立最終備份。

### 建立最終備份 (主控台)

當您使用主控台刪除 Memcached 無伺服器快取時，您可以建立最終備份。ElastiCache

若要在刪除快取時建立最終備份，請在刪除對話方塊中選擇 [建立備份] 下的 [是]，並為備份命名。

### 相關主題

- [使用 AWS Management Console](#)

### 建立最終備份 (AWS CLI)

您可以在使用刪除快取時建立最終備份 AWS CLI。

### 主題

- [刪除無伺服器快取時](#)

### 刪除無伺服器快取時

若要建立最終備份，請使用具有下列參數的 delete-serverless-cache AWS CLI 作業。

- --serverless-cache-name - 要刪除的快取名稱。
- --final-snapshot-name - 備份的名稱。

下列程式碼會在刪除快取 myserverlesscache 時建立最終備份 bkup-20231127-final。

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-serverless-cache \  
    --serverless-cache-name myserverlesscache \  
    --final-snapshot-name bkup-20231127-final
```

針對 Windows：

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

```
--serverless-cache-name myserverlesscache ^  
--final-snapshot-name bkup-20231127-final
```

如需詳細資訊，請參閱《AWS CLI 命令參考》中的 [delete-serverless-cache](#)。

## 描述備份

下列程序說明如何顯示您的備份清單。如果您需要，也可以檢視特定備份的詳細資訊。

### 描述備份 (主控台)

若要使用顯示備份 AWS Management Console

1. 請登入 AWS Management Console 並開啟 ElastiCache 主控台，網址為 <https://console.aws.amazon.com/elasticache/>。
2. 從導覽窗格，選擇 Backups (備份)。
3. 若要查看特定備份的詳細資訊，請選擇備份名稱左側的方塊。

### 說明無伺服器備份 (AWS CLI)

若要顯示無伺服器備份的清單，以及選擇性地顯示特定備份的詳細資訊，請使用 `describe-serverless-cache-snapshots` CLI 操作。

### 範例

下列操作使用參數 `--max-records` 列出最多 20 個與您的帳戶相關的備份。省略參數 `--max-records` 會列出最多 50 個備份。

```
aws elasticache describe-serverless-cache-snapshots --max-records 20
```

下列操作使用參數 `--serverless-cache-name` 單獨列出與快取 `my-cache` 相關聯的備份。

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-name my-cache
```

下列操作使用參數 `--serverless-cache-snapshot-name` 顯示備份 `my-backup` 的詳細資訊。

```
aws elasticache describe-serverless-cache-snapshots --serverless-cache-snapshot-name my-backup
```

如需詳細資訊，請參閱 [《命令參考》中描述的無伺服器快取快照集](#)。AWS CLI

## 複製備份

您可以建立任何備份的副本，不論是自動或手動建立的備份。

下列步驟說明如何複製備份。

複製備份 (主控台)

複製備份 (主控台)

1. 請登入 AWS Management Console 並開啟 ElastiCache 主控台，網址為 <https://console.aws.amazon.com/elasticache/>。
2. 若要查看您的備份清單，請從左側導覽窗格，選擇 Backups (備份)。
3. 從備份清單，選擇您要複製之備份名稱左側的方塊。
4. 選擇動作、複製。
5. 在 New backup name (新的備份名稱) 方塊中，輸入新的備份名稱。
6. 請選擇 Copy (複製)。

複製無伺服器備份 (AWS CLI)

若要複製無伺服器快取的備份，請使用 `copy-serverless-cache-snapshot` 操作。

參數

- `--source-serverless-cache-snapshot-name` - 要複製的備份名稱。
- `--target-serverless-cache-snapshot-name` - 備份的複本名稱。

下列範例會建立自動備份的複本。

若為 Linux、macOS 或 Unix：

```
aws elasticache copy-serverless-cache-snapshot \  
  --source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 \  
  --target-serverless-cache-snapshot-name my-backup-copy
```

針對 Windows：

```
aws elasticache copy-serverless-cache-snapshot ^
```

```
--source-serverless-cache-snapshot-name automatic.my-cache-2023-11-27-03-15 ^  
--target-serverless-cache-snapshot-name my-backup-copy
```

如需詳細資訊，請參閱 AWS CLI 中的 [copy-serverless-cache-snapshot](#)。

## 從備份還原到新的快取

您可以將現有備份還原到新的無伺服器快取叢集中。

將備份還原到無伺服器快取 (主控台)

將備份還原到無伺服器快取 (主控台)

1. 請登入 AWS Management Console 並開啟 ElastiCache 主控台，[網址為 https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/)。
2. 從導覽窗格，選擇 Backups (備份)。
3. 在備份清單中，選擇您要還原之備份名稱左側的方塊。
4. 選擇動作，然後選擇還原。
5. 輸入新的無伺服器快取的名稱，以及選用的說明。
6. 按一下建立以建立新的快取，並從備份匯入資料。

將備份還原到無伺服器快取 (AWS CLI)

將備份還原到新的無伺服器快取 (AWS CLI)

下列 AWS CLI 範例會使用建立新的快取，`create-serverless-cache`並從備份匯入資料。

若為 Linux、macOS 或 Unix：

針對 Windows：

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached \  
  --snapshot-arns-to-restore Snapshot-ARN
```

針對 Windows：

```
aws elasticache create-serverless-cache ^ \  
  --serverless-cache-name CacheName ^ \  
  --engine memcached ^ \  
  --snapshot-arns-to-restore Snapshot-ARN
```

## 刪除備份

自動備份會在超過保留期限之後自動刪除。如果您刪除叢集，也會刪除其所有自動備份。如果您刪除複寫群組，也會從該群組中的叢集刪除所有自動備份。

ElastiCache 提供刪除 API 作業，可讓您隨時刪除備份，無論備份是自動還是手動建立。由於手動備份沒有留存期限，因此手動刪除是唯一的移除方式。

您可以使用 ElastiCache 控制台、AWS CLI、或 ElastiCache API 刪除備份。

### 刪除備份 (主控台)

下列程序會使用 ElastiCache 主控台刪除備份。

#### 刪除備份

1. 請登入 AWS Management Console 並開啟 ElastiCache 主控台，網址為 <https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格中，選擇備份。

Backups (備份) 畫面會隨即顯示，其中包含您的備份清單。

3. 選擇您要刪除之備份名稱左側的方塊。
4. 選擇刪除。
5. 如果要刪除此備份，請在 Delete Backup (刪除備份) 確認畫面上選擇 Delete (刪除)。狀態會變更為「刪除中」。

### 刪除無伺服器備份 (AWS CLI)

使用刪除快照 AWS CLI 作業搭配下列參數來刪除無伺服器備份。

- `--serverless-cache-snapshot-name` - 要刪除的備份名稱。

下列程式碼會刪除備份 `myBackup`。

```
aws elasticache delete-serverless-cache-snapshot --serverless-cache-snapshot-name myBackup
```

如需詳細資訊，請參閱《AWS CLI 命令參考》中的 [delete-serverless-cache-snapshot](#)。

## 標記備份

您可以用標籤的形式將自己的中繼資料指派給每個備份。標籤可讓您以不同的方式分類備份，例如依據目的、擁有者或環境。當您有許多相同類型的資源時，這將會很有用，因為—您可以依據先前指派的標籤，快速識別特定的資源。如需詳細資訊，請參閱 [您可以標記的資源](#)。

成本分配標籤是透過標籤值將發票上的費用分組，藉此在多個 AWS 服務中追蹤成本的一種方式。若要進一步了解成本分配標籤，請參閱 [使用成本分配標籤](#)。

使用 ElastiCache 主控台、或 ElastiCache API AWS CLI，您可以在備份上新增、列出、修改、移除或複製成本配置標籤。如需更多詳細資訊，請參閱 [使用成本配置標籤監控成本](#)。

# 引擎版本和升級

本節涵蓋支援的 Memcached 引擎版本以及升級方法。

## 主題

- [支援的 ElastiCache for Memcached 版本](#)
- [引擎版本和升級](#)
- [如何升級引擎版本](#)

## 支援的 ElastiCache for Memcached 版本

ElastiCache 支援下列 Memcached 版本，且可升級為更新版本。升級至較新版本時，請特別注意必須滿足的相關條件，以免升級失敗。

### ElastiCache for Memcached 版本

- [Memcached 1.6.22 版](#)
- [Memcached 1.6.17 版](#)
- [Memcached 1.6.12 版](#)
- [Memcached 1.6.6 版](#)
- [Memcached 1.5.16 版](#)
- [Memcached 1.5.10 版](#)
- [Memcached 1.4.34 版](#)
- [Memcached 1.4.33 版](#)
- [Memcached 1.4.24 版](#)
- [Memcached 1.4.14 版](#)
- [Memcached 1.4.5 版](#)

### Memcached 1.6.22 版

ElastiCache for Memcached 新增 Memcached 1.6.22 版支援。其中沒有任何新功能，但包括錯誤修正以及 [Memcached 1.6.18](#) 的累積更新。

如需詳細資訊，請參閱 GitHub 中 Memcached 上的 [ReleaseNotes1622](#)。

### Memcached 1.6.17 版

ElastiCache for Memcached 新增 Memcached 1.6.17 版支援。其中沒有任何新功能，但包括錯誤修正以及 [Memcached 1.6.17](#) 的累積更新。

如需詳細資訊，請參閱 GitHub 中 Memcached 頁面上的 [ReleaseNotes1617](#)。

### Memcached 1.6.12 版

ElastiCache for Memcached 新增 Memcached 1.6.12 版與傳輸中加密的支援。其中包括 [Memcached 1.6.6 版](#) 的錯誤修正與累積更新。

如需詳細資訊，請參閱 GitHub 中 Memcached 頁面上的 [ReleaseNotes1612](#)。

## Memcached 1.6.6 版

ElastiCache for Memcached 新增 Memcached 1.6.6 版支援。其中沒有任何新功能，但包括錯誤修正以及 [Memcached 1.5.16](#) 的累積更新。ElastiCache for Memcached 不包含 [Extstore](#)。

如需詳細資訊，請參閱 GitHub 中 Memcached 頁面上的 [ReleaseNotes166](#)。

## Memcached 1.5.16 版

ElastiCache for Memcached 新增 Memcached 1.5.16 版支援。其中沒有任何新功能，但包括錯誤修正以及 [Memcached 1.5.14](#) 與 [Memcached 1.5.15](#) 的累積更新。

如需詳細資訊，請參閱 GitHub 中 Memcached 頁面的 [Memcached 1.5.16 版本備註](#)。

## Memcached 1.5.10 版

ElastiCache for Memcached 1.5.10 版支援下列 Memcached 功能：

- 自動化 slab 重新平衡。
- 使用 murmur3 演算法進行更快速的雜湊資料表查閱。
- 分段的 LRU 演算法。
- LRU 爬蟲程式用來在背景回收記憶體。
- `--enable-seccomp`：編譯時間選項。

它也推出 `no_modern` 和 `inline_ascii_resp` 方法。如需更多詳細資訊，請參閱 [Memcached 1.5.10 參數變更](#)。

從適用於 Memcached 1.4.34 版的 ElastiCache 起新增的改良功能，包括下列項目：

- 累積修正程式，例如 ASCII multiget、CVE-2017-9951 和 metadumper 的限制爬蟲程式。
- 在達到連線限制時關閉連線來獲得更好的連線管理。
- 對於 1MB 以上的項目大小，改善的項目大小管理。
- 透過將每一項目的記憶體需求減少一些位元組，獲得更好的效能和記憶體負擔改善。

如需詳細資訊，請參閱 GitHub 中 Memcached 頁面的 [Memcached 1.5.10 版本備註](#)。

## Memcached 1.4.34 版

ElastiCache for Memcached 1.4.34 版並無新增任何 1.4.33 版沒有的新功能。1.4.34 版是一款錯誤修正版本，其比一般類似版本的涵蓋範圍更大。

如需詳細資訊，請參閱 GitHub 中 Memcached 頁面的 [Memcached 1.4.34 版本備註](#)。

## Memcached 1.4.33 版

從 Memcached 1.4.24 版起新增的改良功能，包括下列項目：

- 可傾印特定 slab 類別、slab 類別清單或所有 slab 類別的所有中繼資料。如需詳細資訊，請參閱 [Memcached 1.4.31 版本備註](#)。
- 已改善對預設超過 1 megabyte (MB) 大型項目的支援。如需詳細資訊，請參閱 [Memcached 1.4.29 版本備註](#)。
- 可指定在詢問關閉用戶端之前用戶端可閒置的時間。

可動態增加 Memcached 可用的記憶體容量，而不需重新啟動叢集。如需詳細資訊，請參閱 [Memcached 1.4.27 版本備註](#)。

- 現已支援 fetchers、mutations 和 evictions 的記錄功能。如需詳細資訊，請參閱 [Memcached 1.4.26 版本備註](#)。
- 釋放的記憶體可以回收到全域集區，並重新指派給新的 slab 類別。如需詳細資訊，請參閱 [Memcached 1.4.25 版本備註](#)。
- 多項錯誤修正。
- 一些新的命令和參數。如需清單，請參閱 [Memcached 1.4.33 版新增參數](#)。

## Memcached 1.4.24 版

從 Memcached 1.4.14 版起新增的改良功能，包括下列項目：

- 使用背景程序來進行最久未使用的項目 (LRU) 管理。
- 已新增將 jenkins 或 murmur3 做為雜湊演算法的選項。
- 一些新的命令和參數。如需清單，請參閱 [Memcached 1.4.24 版新增參數](#)。
- 多項錯誤修正。

## Memcached 1.4.14 版

從 Memcached 1.4.5 版起新增的改良功能，包括下列項目：

- 增強型 slab 重新平衡功能。
- 高效能與可擴展性提升。
- 引進 touch 命令，以更新現有項目的過期時間，而不需擷取該項目。
- 自動探索 - 可讓用戶端程式自動判斷快取叢集中的所有快取節點，並初始化及維持與這些節點的連線。

## Memcached 1.4.5 版

Memcached 1.4.5 版是 Amazon ElastiCache for Memcached 支援的初始引擎與版本。

## 引擎版本和升級

MAJOR 版本適用於 API 不相容變更，而 MINOR 版本適用於以回溯相容方式新增的功能。PATCH 版本適用於回溯相容的錯誤修正和非功能性變更。

### ElastiCache Serverless 的版本管理

ElastiCache Serverless 會自動將最新的次要和修補程式軟體版本套用至您的快取，而不會對應用程式造成任何影響或停機。您不需要進行任何動作。

當推出新的主要版本時，ElastiCache Serverless 會在主控台傳送通知給您，並在 EventBridge 中傳送事件。您可以選擇使用主控台、CLI 或 API 修改快取，然後選取最新的引擎版本，以將快取升級至最新的主要版本。

### 自行設計的 ElastiCache 叢集的版本管理

使用自行設計的 ElastiCache 叢集時，您可以控制何時將支援您快取叢集的軟體升級到 ElastiCache 支援的新版本。您可以控制何時將快取升級到最新可用的主要、次要和修補程式版本。您可以修改引擎版本並指定新的引擎版本，以初始化叢集或複寫群組的引擎版本升級。

您可以控制是否以及何時將支援快取叢集的通訊協定合規軟體升級到 ElastiCache 支援的新版本。這一層控制可讓您維持特定版本的相容性、在部署至生產環境前先利用您的應用程式測試新版本，並根據自己的期限和時間表執行版本升級。

由於版本升級可能涉及一定的相容性風險，因此這些升級不會自行啟動，必須由您啟動。

若要升級至更新版本的 Memcached，請指定您要使用的新引擎版本，以修改快取叢集。升級至更新版本的 Memcached 是一種破壞性程序，您會失去所有資料，並從空白快取開始。如需詳細資訊，請參閱 [修改叢集](#)。

從舊版 Memcached 升級到 Memcached 1.4.33 版或更新版本時，請注意下列要求。CreateCacheCluster 和 ModifyCacheCluster 必須滿足下列條件：

- 如果  $\text{slab\_chunk\_max} > \text{max\_item\_size}$ 。
- 如果  $\text{max\_item\_size} \bmod \text{slab\_chunk\_max} \neq 0$ 。
- 如果  $\text{max\_item\_size} > ((\text{max\_cache\_memory} - \text{memcached\_connections\_overhead}) / 4)$ 。

$(\text{max\_cache\_memory} - \text{memcached\_connections\_overhead})$  值是節點可供資料使用的記憶體。如需更多詳細資訊，請參閱 [Memcached 連線額外負荷](#)。

## 使用自行設計的叢集時的升級考量

### Note

下列考量僅適用於升級自行設計之叢集的情況。不適用 ElastiCache Serverless。

升級自行設計的叢集時，請考量下列事項：

- 引擎版本管理功能是為了讓您能夠盡可能控制執行修補的方式，但在系統或快取軟體可能存在關鍵安全漏洞的情況下，ElastiCache 保留代您修補叢集的權利。
- 由於 Memcached 引擎不支援永久儲存，因此 Memcached 引擎版本升級一定是破壞性程序，並會清除叢集中的所有快取資料。

### 如何升級引擎版本

若要啟動叢集的版本升級，請對其進行修改並指定較新的引擎版本。您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 進行升級：

- 若要使用 AWS Management Console，請參閱 - [透過主控台修改叢集](#)。
- 若要使用 AWS CLI，請參閱[使用 CLI 修改叢集](#)。
- 若要使用 ElastiCache API，請參閱[透過 API 修改叢集](#)。

## 如何升級引擎版本

若要啟動叢集的版本升級，請對其進行修改並指定較新的引擎版本。您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 進行升級：

- 若要使用 AWS Management Console，請參閱「[使用 AWS Management Console](#)」。
- 若要使用 AWS CLI，請參閱[使用 AWS CLI](#)。
- 若要使用 ElastiCache API，請參閱「[應用 ElastiCache 程式介面](#)」。

## ElastiCache 最佳實踐和緩存策略

您可以在下面找到適用於 Amazon 的建議最佳實務 ElastiCache。遵循這些內容可改善快取的效能和可靠性。

### 主題

- [Memcached 用戶端的最佳實務](#)
- [支援的 Memcached 命令](#)
- [快取策略](#)

## Memcached 用戶端的最佳實務

若要進一步了解使用常用的開放原始碼 Memcached 用戶端程式庫與 ElastiCache 資源互動的最佳實務，請參閱下列主題。

### 主題

- [設定您的 ElastiCache 用戶端以達到高效率的負載平衡](#)
- [IPv6 用戶端範例](#)

## 設定您的 ElastiCache 用戶端以達到高效率的負載平衡

### Note

本節適用於自行設計的多節點 Memcached 叢集。

若要有效使用多個 ElastiCache Memcached 節點，您需要將快取索引鍵分散到各個節點。其中一種針對擁有  $n$  個節點叢集進行負載平衡的簡易方式，便是計算物件索引鍵的雜湊，並以  $n$  對結果進行模除以取得餘數 -  $\text{hash}(\text{key}) \bmod n$ 。產生的值 (0 到  $n-1$ ) 便是您置放物件的節點數。

只要節點數 ( $n$ ) 為常數，此方法便相當簡易且運作良好。但是，每當您新增節點到叢集，或從叢集移除節點時，需要移動的鍵數便是  $(n - 1) / n$  (其中  $n$  是新的節點數)。因此，此方法會導致移動大量的鍵，引發大量的初始快取遺漏，尤其是在節點數越來越大時。從 1 個擴展到 2 個節點會導致  $(2-1)/2$  (50%) 的索引鍵移動，這是最佳情況。從 9 個擴展到 10 個節點會導致  $(10-1)/10$  (90%) 的索引鍵移動。若您因為流量的峰值而向上擴展，您不會希望發生大量的快取遺漏。大量的快取遺漏會導致資料庫遭受衝擊，尤其是藉時資料庫已經因為流量峰值而超載了。

此難題的解決方案便是一致性雜湊。一致雜湊所使用的演算法，會讓節點新增到叢集或從叢集移除節點時需移動的索引鍵數約為  $1/n$  (其中  $n$  是新節點的數量)。從 1 擴展到 2 個節點會導致  $1/2$  (50%) 的鍵移動，此為最壞情況。從 9 擴展到 10 個節點會導致  $1/10$  (10%) 的鍵移動。

做為使用者，您可以控制要用於多節點叢集的雜湊演算法。建議您設定用戶端，以使用一致性雜湊。所幸，在大多數的熱門語言中，有許多 Memcached 用戶端程式庫都實作了一致性雜湊。檢查您使用的程式庫文件，查看其是否支援一致性雜湊以及其實作方式。

若您使用 Java、PHP 或 .NET，建議您使用其中一個 Amazon ElastiCache 用戶端程式庫。

### 使用 Java 的一致性雜湊

ElastiCache Memcached Java 用戶端是以開放原始碼的 `spymemcached` Java 用戶端為基礎，其內建了一致雜湊功能。程式庫包含實作一致性雜湊的 `KetamaConnectionFactory` 類別。根據預設，`spymemcached` 中會關閉一致性雜湊。

如需詳細資訊，請參閱 [KetamaConnectionFactory](#) 中的 `KetamaConnectionFactory` 文件。

### 使用 PHP 的一致雜湊

ElastiCache Memcached PHP 用戶端是內建 Memcached PHP 程式庫的包裝函數。根據預設，Memcached PHP 程式庫會關閉一致性雜湊。

請使用以下程式碼來開啟一致性雜湊。

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

除了上述程式碼之外，我們建議也在您的 `php.ini` 檔案中開啟 `memcached.sess_consistent_hash`。

如需詳細資訊，請參閱位於 <http://php.net/manual/en/memcached.configuration.php> 的 Memcached PHP 執行時間設定文件。請特別注意 `memcached.sess_consistent_hash` 參數。

使用 .NET 的一致雜湊

ElastiCache Memcached .NET 用戶端是 Enyim Memcached 的包裝函數。根據預設，Enyim Memcached 用戶端會開啟一致性雜湊。

如需詳細資訊，請參閱 `memcached/locator` 說明文件，網址為 <https://github.com/enyim/EnyimMemcached/wiki/MemcachedClient-Configuration#user-content-memcachedlocator>。

## IPv6 用戶端範例

### Note

本節適用於自行設計的 Memcached 叢集。

ElastiCache 與開源內存兼容。這表示支援 IPv6 連線的 Memcached 開放原始碼用戶端應該能夠連線到已啟用記憶體快取叢集 ElastiCache 的 IPv6。此外，下列用戶端已經過特別驗證，可與所有受支援的網路類型組態搭配使用：

以下是與常用的開源客戶端庫 IPv6 啟用 ElastiCache 資源進行交互的最佳實踐。您可以檢視[與互動的現有最佳做法，以 ElastiCache](#)取得設定 ElastiCache 資源用戶端的建議。但是，與啟用 IPv6 的資源互動時，需注意一些事項。

已驗證的用戶端

已驗證的用戶端：

- [AWS ElastiCache 適用於 PHP 的叢集用戶端記憶體快取 — 版本 \\*3.6.2](#)
- [AWS ElastiCache Java 的集群客戶端內存緩存 — Github 上的最新主機](#)

設定雙堆疊叢集的偏好通訊協定

針對 Memcached 叢集，您可以使用 IP 探索參數，控制用戶端用來連線至叢集中節點的通訊協定。IP 探索參數可以設為 IPv4 或 IPv6。

IP 探索參數可控制設定取得叢集輸出中使用的 IP 通訊協定。這些通訊協定將決定用戶端所使用的 IP 通訊協定，這些通訊協定支援 Memcached 叢集的 ElastiCache 自動探索功能。

變更 IP 探索不會導致連線用戶端停機。但是，該變更需要一些傳播時間。

監控 Java 的 `getAvailableNodeEndpoints` 輸出，針對 Php 則監控 `getServerList` 的輸出。一旦這些函數的輸出報告了叢集中使用更新通訊協定的所有節點 IP，就表示變更已完成傳播。

Java 範例：

```
MemcachedClient client = new MemcachedClient(new InetSocketAddress("xxxx", 11211));

Class targetProtocolType = Inet6Address.class; // Or Inet4Address.class if you're
switching to IPv4

Set<String> nodes;

do {
    nodes =
    client.getAvailableNodeEndpoints().stream().map(NodeEndPoint::getIpAddress).collect(Collectors.toList());

    Thread.sleep(1000);
} while (!nodes.stream().allMatch(node -> {
    try {
        return finalTargetProtocolType.isInstance(InetAddress.getByAddress(node));
    } catch (UnknownHostException ignored) {}
    return false;
})));
```

Php 範例：

```
$client = new Memcached;
$client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);
$client->addServer("xxxx", 11211);

$nodes = [];
$target_ips_count = 0;
do {
    # The PHP memcached client only updates the server list if the polling interval has
    expired and a
    # command is sent
    $client->get('test');

    $nodes = $client->getServerList();

    sleep(1);
```

```
$target_ips_count = 0;

// For IPv4 use FILTER_FLAG_IPV4
$target_ips_count = count(array_filter($nodes, function($node) { return
filter_var($node["ipaddress"], FILTER_VALIDATE_IP, FILTER_FLAG_IPV6); }));

} while (count($nodes) !== $target_ips_count);
```

在更新 IP 探索之前建立的任何現有用戶端連線仍會使用舊通訊協定進行連線。一旦在叢集探索命令的輸出中偵測到變更，所有經過驗證的用戶端都會使用新的 IP 通訊協定自動重新連線至叢集。但是，這仍取決於用戶端的操作。

## 啟用 TLS 的雙堆疊 ElastiCache 叢集

為叢集啟用 ElastiCache TLS 時，叢集探查功能會傳 `config get cluster` 回主機名稱而非 IP。然後使用主機名稱而不是 IP 來連接到 ElastiCache 叢集並執行 TLS 交握。這表示用戶端不會受到 IP 探索參數的影響。針對啟用 TLS 的叢集，IP 探索參數對偏好的 IP 通訊協定沒有影響。反之，用戶端在解析 DNS 主機名稱時，偏好使用的 IP 通訊協定會決定所用的 IP 通訊協定。

## Java 用戶端

從同時支援 IPv4 和 IPv6 的 Java 環境連線時，為了回溯相容性，Java 會預設偏好 IPv4 而非 IPv6。不過，IP 通訊協定偏好設定可透過 JVM 引數設定。為了偏好 IPv4，JVM 會接受 `-Djava.net.preferIPv4Stack=true` 並偏好 IPv6 設為 `-Djava.net.preferIPv6Stack=true`。設定 `-Djava.net.preferIPv4Stack=true` 就表示 JVM 將不再進行任何 IPv6 連線。

## 主機層級偏好

一般而言，若用戶端或用戶端執行期未提供設定 IP 通訊協定偏好設定的組態選項，在執行 DNS 解析時，IP 通訊協定將取決於主機的組態。根據預設，大部份主機偏好 IPv6 而非 IPv4，但可以在主機層級設定此偏好設定。這將影響來自該主機的所有 DNS 請求，而不僅僅是那些對 ElastiCache 叢集的 DNS 請求。

## Linux 主機

針對 Linux，可以透過修改 `gai.conf` 檔案，設定 IP 通訊協定偏好設定。您可以在 `/etc/gai.conf` 下找到 `gai.conf` 檔案。若沒有指定的 `gai.conf`，則應在 `/usr/share/doc/glibc-common-x.xx/gai.conf` 之下提供一個範例，可以複製到 `/etc/gai.conf`，且預設組態應取消註解。若要在連線至叢集時將組態更新為偏好 IPv4，請更新包含 ElastiCache 叢集 IP 的 CIDR 範圍的優先順序，

使其高於預設 IPv6 連線的優先順序。根據預設，IPv6 連線的優先順序為 40。例如，若該叢集位於具有 CIDR 172.31.0.0/16 的子網路中，以下組態會導致用戶端在連線至叢集時偏好 IPv4。

```
label ::1/128      0
label ::/0        1
label 2002::/16   2
label ::/96       3
label ::ffff:0:0/96 4
label fec0::/10   5
label fc00::/7    6
label 2001:0::/32 7
label ::ffff:172.31.0.0/112 8
#
# This default differs from the tables given in RFC 3484 by handling
# (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
# The reason for this difference is that these addresses are never
# NATed while IPv4 site-local addresses most probably are. Given
# the precedence of IPv6 over IPv4 (see below) on machines having only
# site-local IPv4 and IPv6 addresses a lookup for a global address would
# see the IPv6 be preferred. The result is a long delay because the
# site-local IPv6 addresses cannot be used while the IPv4 address is
# (at least for the foreseeable future) NATed. We also treat Teredo
# tunnels special.
#
# precedence <mask> <value>
# Add another rule to the RFC 3484 precedence table. See section 2.1
# and 10.3 in RFC 3484. The default is:
#
precedence ::1/128      50
precedence ::/0        40
precedence 2002::/16   30
precedence ::/96       20
precedence ::ffff:0:0/96 10
precedence ::ffff:172.31.0.0/112 100
```

若需更多 `gai.conf` 的相關資訊，請前往 [Linux 主頁](#)

## Windows 主機

Windows 主機的程式十分類似。針對 Windows 主機，您可以執行 `netsh interface ipv6 set prefix CIDR_CONTAINING_CLUSTER_IPS PRECEDENCE LABEL`。此操作與在 Linux 主機上修改 `gai.conf` 檔案的效果相同。

這會更新偏好設定政策，使其在指定的 CIDR 範圍內偏好 IPv4 連線而非 IPv6 連線。例如，若叢集位於具執行 `netsh interface ipv6 set prefix ::ffff:172.31.0.0:0/112 100 15` 的 172.31.0.0:0/16 CIDR 子網路中，則會產生下列優先順序表格，導致用戶端在連線至叢集時偏好 IPv4。

```
C:\Users\Administrator>netsh interface ipv6 show prefixpolicies
Querying active state...

Precedence Label Prefix
-----
100 15 ::ffff:172.31.0.0:0/112
20 4 ::ffff:0:0/96
50 0 ::1/128
40 1 ::/0
30 2 2002::/16
5 5 2001::/32
3 13 fc00::/7
1 11 fec0::/10
1 12 3ffe::/16
1 3 ::/96
```

## 支援的 Memcached 命令

ElastiCache 適用於 Memcached 的無伺服器支援開放原始碼 memcached 1.6 中的所有 Memcached [命令](#)，但下列項目除外：

- 用戶端連線需要 TLS，因此不支援 UDP 通訊協定。
- 不支援二進位通訊協定，因為已在 memcached 1.6 中正式[棄用](#)。
- GET/GETS 命令限制為 16KB，以避免可能遭到對伺服器進行大量金鑰擷取的 DoS 攻擊。
- 延遲的 flush\_all 命令將遭到拒絕，並顯示 CLIENT\_ERROR。
- 不支援設定引擎或揭露有關引擎狀態或日誌之內部資訊的命令，例如：
  - 對於 STATS 命令，僅支援 stats 和 stats reset。其他變化都將傳回 ERROR
  - lru / lru\_crawler - 修改 LRU 和 LRU 爬蟲程式設定
  - watch - 監看 memcached 伺服器日誌
  - verbosity - 設定伺服器日誌層級
  - me-不支持元調試 ( me ) 命令

## 快取策略

下列主題提供填入和維護快取的策略。

您實作用於填入和維護快取的策略，取決於您要快取的資料，以及該資料的存取模式。例如，您可能不會想對遊戲網站前十名排行榜和趨勢新聞報導使用相同的策略。在本節的其餘部分，我們會討論常見的快取維護策略，以及其優點和缺點。

### 主題

- [延遲載入](#)
- [全部寫入](#)
- [新增 TTL](#)
- [相關主題](#)

## 延遲載入

如同名稱所隱含的意義，延遲載入是一項快取策略，只有在必要時才會將資料載入快取中。它的運作方式如下。

Amazon ElastiCache 是記憶體內鍵值存放區，位於應用程式與其存取的資料存放區 (資料庫) 之間。每當應用程式請求資料，它會先向 ElastiCache 快取進行請求。如果資料存在於快取且是最新的，ElastiCache 會將該資料傳回至您的應用程式。如果資料不存在於快取中或已過期，則應用程式會向資料存放區請求資料。接著資料存放區會將資料傳回至應用程式。應用程式會將從存放區收到的資料寫入快取中。如此一來，下次收到請求時就能更快速擷取這些資料。

當資料存在於快取中且未過期，就會發生快取命中：

1. 應用程式向快取請求資料。
2. 快取將資料傳回給應用程式。

當資料不存在於快取中已過期，就會發生快取未命中：

1. 應用程式向快取請求資料。
2. 快取中沒有所請求的資料，因此傳回 null。
3. 應用程式請求並收到來自資料庫的資料。
4. 應用程式以新資料更新快取。

## 延遲載入的優點和缺點

延遲載入的優點如下：

- 只會將請求的資料加以快取。

因為大部分資料永遠不會被請求，延遲載入可避免將快取填滿未請求的資料。

- 節點故障不會成為應用程式的嚴重問題。

當節點故障並且由新的空白節點取代時，應用程式會繼續運作，只是會增加延遲。對新節點提出請求時，每次快取未命中都會產生一個資料庫查詢。同時會將資料複本新增至快取中，以便從快取擷取後續的請求。

延遲載入的缺點如下：

- 快取遺漏會有懲罰。每次快取未命中都會產生 3 趟行程：

1. 向快取初始請求資料
2. 查詢資料庫來取得資料
3. 將資料寫入快取

這些未命中可能會造成應用程式取得資料的時間出現明顯的延遲。

- 資料過時。

如果只在快取未命中時將資料寫入快取，則快取中的資料可能變得過時。當資料庫中的資料變更時，並不會更新快取，因此會產生此結果。若要解決這個問題，您可以使用 [全部寫入](#) 和 [新增 TTL](#) 策略。

## 延遲載入虛擬程式碼範例

以下是延遲載入邏輯的虛擬程式碼範例。

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

在此範例中，取得資料的應用程式程式碼如下。

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

## 全部寫入

每當有資料寫入資料庫，全部寫入 (write-through) 策略就會在快取中新增或更新資料。

### 全部寫入的優點和缺點

全部寫入的優點如下：

- 快取中的資料絕不會過時。

快取中的資料每次寫入資料庫時都會進行更新，因此快取中的資料永遠是最新的。

- 寫入懲罰與讀取懲罰。

每次寫入都牽涉到兩個來回行程：

1. 一個寫入快取
2. 一個寫入資料庫

這會對程序增加延遲。也就是說，相較於擷取資料，最終使用者一般更能容忍更新資料時的延遲。固有的想法是更新需要的動作較多因此會花費較長時間。

全部寫入的缺點如下：

- 遺漏資料。

如果啟動新節點，無論是因為節點故障或水平擴展，都會發生資料未命中。這些資料會繼續處於未命中狀態，直到在資料庫上新增或更新為止。您可以藉由搭配全部寫入實作[延遲載入](#)來盡可能減少此情形。

- 快取流失。

大部分資料都永遠不會被讀取，導致資源的浪費。您可以藉由[新增存留時間 \(TTL\) 值](#)，將浪費的空間降到最少。

## 全部寫入虛擬程式碼範例

下列是全部寫入邏輯的虛擬程式碼範例。

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

在此範例中，取得資料的應用程式程式碼如下。

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

## 新增 TTL

延遲載入允許使用過時資料，但不會因空白節點而失敗。全部寫入可確保資料總是保持最新狀態，但可能因為空白節點而失敗，且可能對快取填入多餘的資料。對每個寫入新增存留時間 (TTL) 值，便可享受每個策略的優點。同時可以大幅避免將快取塞滿多餘的資料。

存留時間 (TTL) 是整數值，指定索引鍵到期之前的秒數。Memcached 需以秒為單位指定此值。應用程式嘗試讀取過期的索引鍵時，系統會視為找不到索引鍵。資料庫會查詢索引鍵並更新快取。這種方法並不能保證值不會過時。不過會確保資料不致於過時太久，且需要偶爾從資料庫重新整理快取中的值。

如需詳細資訊，請參閱 [Memcached set 命令](#)。

## TTL 虛擬程式碼

下列程式碼是使用 TTL 的全部寫入邏輯的虛擬程式碼範例。

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

下列程式碼是使用 TTL 的延遲載入邏輯的虛擬程式碼範例。

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

在此範例中，取得資料的應用程式程式碼如下。

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

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

## 相關主題

- [記憶體內資料存放區](#)
- [選擇引擎和版本](#)
- [縮放記憶 ElastiCache 體快取](#)

## 管理您的自行設計叢集

本節包含可協助您管理自行設計叢集的主題。

### Note

這些主題不適用 ElastiCache Serverless。

## 主題

- [管理維護作業](#)
- [使用參數群組設定引擎參數](#)

## 管理維護作業

每個叢集都有每週一次的維護時段，會在此期間套用任何系統變更。如果您在建立或修改叢集時未指定偏好的維護時段，請在您區域的維護時段內，在一週中隨機選擇的日期指 ElastiCache 派 60 分鐘的維護時段。

60 分鐘的維護時段隨機選自每個區域的 8 小時時段。以下資料表列出每個區域的時段，預設維護時段會從此時段中指派。您可以選擇區域維護時段以外的偏好維護時段。

區域代碼	區域名稱	區域維護時段
ap-northeast-1	亞太 (東京) 區域	13:00–21:00 UTC
ap-northeast-2	亞太 (首爾) 區域	12:00–20:00 UTC
ap-northeast-3	亞太 (大阪) 區域	12:00–20:00 UTC
ap-southeast-3	亞太區域 (雅加達)	14:00–22:00 UTC
ap-south-1	亞太 (孟買) 區域	17:30–1:30 (UTC)
ap-southeast-1	亞太 (新加坡) 區域	14:00–22:00 UTC
cn-north-1	中國 (北京) 區域	下午 2 時至 10 時 (UTC)
cn-northwest-1	中國 (寧夏) 區域	14:00–22:00 UTC
ap-east-1	亞太區域 (香港) 區域	13:00–21:00 UTC
ap-southeast-2	亞太 (雪梨) 區域	12:00–20:00 UTC
eu-west-3	歐洲 (巴黎) 區域	下午 11 時 59 分至上午 7 時 29 分 (UTC)
af-south-1	非洲 (開普敦) 區域	13:00–21:00 UTC
eu-central-1	歐洲 (法蘭克福) 區域	23:00–07:00 UTC
eu-west-1	歐洲 (愛爾蘭) 區域	下午 10 時至次日上午 6 時 (UTC)
eu-west-2	歐洲 (倫敦) 區域	23:00–07:00 UTC
me-south-1	Middle East (Bahrain) Region	13:00–21:00 UTC
me-central-1	中東 (阿拉伯聯合大公國) 區域	13:00–21:00 UTC
eu-south-1	Europe (Milan) Region	21:00–05:00 UTC
sa-east-1	南美洲 (聖保羅) 區域	01:00–09:00 (UTC)

區域代碼	區域名稱	區域維護時段
us-east-1	美國東部 (維吉尼亞北部) 區域	上午 3 時至上午 11 時 (UTC)
us-east-2	美國東部 (俄亥俄) 區域	04:00-12:00 (UTC)
us-gov-west-1	AWS GovCloud (US) 區域	06:00–14:00 UTC
us-west-1	美國西部 (加利佛尼亞北部) 區域	06:00–14:00 UTC
us-west-2	美國西部 (奧勒岡) 區域	06:00–14:00 UTC

### 變更叢集的維護時段

維護時段應落在使用量最低的時段，因此可能需要不時進行調整。您可以修改叢集來指定時間範圍，最多 24 小時，您已請求的所有維護活動會在此期間進行。在此期間會進行所有您請求的延遲或待處理叢集修改。

#### Note

如果您要使用 AWS Management Console [立即套用] 方塊立即套用節點類型修改和/或引擎升級。否則，這些修改將在您排程的下一次維護時段套用。若要使用 API，請參閱[modify-replication-group](#)或[modify-cache-cluster](#)。

### 其他資訊

如需維護時段和節點取代的資訊，請參閱下列內容：

- [ElastiCache 維護 — 維護和節點更換常見問題](#)
- [替換節點](#) - 管理節點更換
- [修改 ElastiCache 叢集](#) - 變更叢集的維護時段

## 使用參數群組設定引擎參數

Amazon ElastiCache 使用參數控制您節點與叢集的執行時間屬性。一般而言，更新的引擎版本會包含額外參數，可支援更新的功能。如需參數表，請參閱 [Memcached 專用參數](#)。

如您所預期的，有些參數值 (例如 `maxmemory`) 會由引擎與節點類型決定。如需根據節點類型的參數值表，請參閱 [Memcached 節點類型專用參數](#)。

 Note

如需 Memcached 限定參數，請參閱 [Memcached 限定參數](#)。

## 主題

- [參數管理](#)
- [快取參數群組層](#)
- [建立參數群組](#)
- [依名稱列出參數群組](#)
- [列出參數群組的值](#)
- [修改參數群組](#)
- [刪除參數群組](#)
- [Memcached 專用參數](#)

## 參數管理

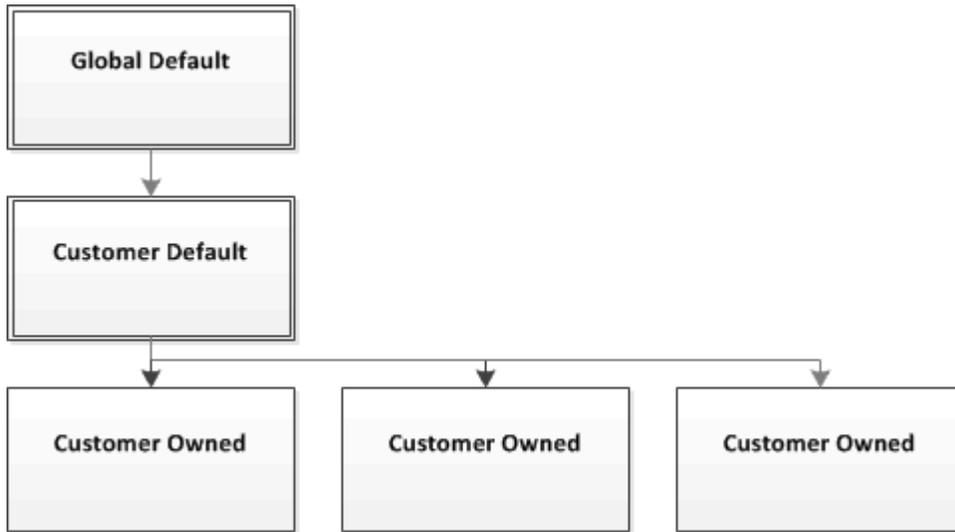
參數會群組成具名參數群組，簡化參數管理。參數群組代表在啟動期間傳遞給引擎軟體的參數特定值組合。這些值會決定每個節點上的引擎程序在執行時間的行為。特定參數群組上的參數值會套用到所有與群組相關聯的節點，無論節點所屬的叢集為何。

若要調整您叢集的效能，您可以修改一部分的參數值，或是變更叢集的參數群組。

- 您無法修改或刪除預設參數群組。若您需要自訂參數值，您必須建立自訂參數群組。
- 參數群組系列及您指派該群組的對象叢集必須相容。例如，若您的叢集執行 Memcached 1.4.8 版，您只能使用來自 Memcached 1.4 系列的參數群組 (預設或自訂)。
- 若您變更叢集的參數群組，任何可條件式修改的參數值在目前及新的參數群組中都必須相同。
- 當您變更叢集的參數時，變更會立即套用至叢集。無論是變更叢集的參數群組本身或是叢集的參數群組內的參數值，均適用此情況。若要判斷特定參數變更何時套用，請參閱 [Memcached 專用參數](#) 資料表中的變更生效。如需重新啟動叢集節點的相關資訊，請參閱 [重新啟動叢集](#)。

## 快取參數群組層

Amazon ElastiCache 有三層快取參數群組，如下所示。



### Amazon ElastiCache 參數群組層

#### 全域預設

區域內所有 Amazon ElastiCache 客戶的最上層根參數群組。

全域預設快取參數群組：

- 會為 ElastiCache 保留，客戶無法使用。

#### 客戶預設

全域預設快取參數群組的複本，其建立的目的是供客戶使用。

客戶預設快取參數群組：

- 由 ElastiCache 建立和擁有。
- 可供客戶用來做為任何執行此快取參數群組所支援引擎版本叢集的快取參數群組使用。
- 客戶無法對其進行編輯。

#### 客戶自有

客戶預設快取參數群組的複本。客戶自有快取參數群組會在客戶建立快取參數群組時建立。

客戶自有快取參數群組：

- 由客戶建立及擁有。
- 可指派給任何客戶的相容叢集。
- 客戶可對其進行修改，建立自訂快取參數群組。

並非所有參數值皆可修改。如需更多詳細資訊，請參閱 [Memcached 專用參數](#)。

## 建立參數群組

若您希望修改不同於預設值的一或多個參數值，便需要建立新的參數群組。您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 來建立參數群組。

### 建立參數群組 (主控台)

下列程序說明如何使用 ElastiCache 主控台建立參數群組。

#### 使用 ElastiCache 主控台建立參數群組

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 若要建立新的參數群組，請選擇 Create Parameter Group (建立參數群組)。

Create Parameter Group (建立參數群組) 畫面隨即出現。

4. 從 Family (系列) 清單，選擇做為您參數群組範本的參數群組系列。

參數群組系列 (例如 memcached1.4) 會在參數群組中定義實際的參數以及其初始值。參數群組系列必須與叢集的引擎和版本一致。

5. 在 Name (名稱) 方塊中，輸入此參數群組的唯一名稱。

建立叢集或修改叢集的參數群組時，您便會根據其名稱選擇參數群組。因此，我們建議選擇附帶資訊且能以某種方式識別參數群組系列的名稱。

參數群組命名限制條件如下：

- 必須以 ASCII 字母開頭。
- 它只能包含 ASCII 字母、數字和連字符號。

- 長度必須介於 1 至 255 個字元之間。
  - 不能連續包含兩個連字號。
  - 結尾不能是連字號。
6. 在 Description (描述) 方塊中，輸入參數群組的描述。
  7. 若要建立參數群組，請選擇 Create (建立)。若要終止程序而不建立參數群組，請選擇 Cancel (取消)。
  8. 建立參數群組時，它會擁有系列的預設值。若要變更預設值，您必須修改參數群組。如需更多詳細資訊，請參閱 [修改參數群組](#)。

### 建立參數群組 (AWS CLI)

若要使用 AWS CLI 建立參數群組，請搭配這些參數，使用 `create-cache-parameter-group` 命令。

- `--cache-parameter-group-name` - 參數群組的名稱。

參數群組命名限制條件如下：

- 必須以 ASCII 字母開頭。
- 它只能包含 ASCII 字母、數字和連字符號。
- 長度必須介於 1 至 255 個字元之間。
- 不能連續包含兩個連字號。
- 結尾不能是連字號。
- `--cache-parameter-group-family` - 參數群組的引擎和版本系列。
- `--description` - 使用者提供的參數群組說明。

### Example

以下範例會建立名為 `myMem14` 的參數群組，並使用 `memcached1.4` 系列做為範本。

若為 Linux、macOS 或 Unix：

```
aws elasticache create-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --cache-parameter-group-family memcached1.4 \  
  --description "My Memcached parameter group"
```

```
--description "My first parameter group"
```

針對 Windows :

```
aws elasticache create-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --cache-parameter-group-family memcached1.4 ^  
  --description "My first parameter group"
```

此命令的輸出看起來會與以下內容相似。

```
{  
  "CacheParameterGroup": {  
    "CacheParameterGroupName": "myMem14",  
    "CacheParameterGroupFamily": "memcached1.4",  
    "Description": "My first parameter group"  
  }  
}
```

建立參數群組時，它會擁有系列的預設值。若要變更預設值，您必須修改參數群組。如需更多詳細資訊，請參閱 [修改參數群組](#)。

如需更多詳細資訊，請參閱 [create-cache-parameter-group](#)。

建立參數群組 (ElastiCache API)

若要使用 ElastiCache API 建立參數群組，請搭配這些參數使用 `CreateCacheParameterGroup` 動作。

- `ParameterGroupName` - 參數群組的名稱。

參數群組命名限制條件如下：

- 必須以 ASCII 字母開頭。
- 它只能包含 ASCII 字母、數字和連字符號。
- 長度必須介於 1 至 255 個字元之間。
- 不能連續包含兩個連字符號。
- 結尾不能是連字符號。
- `CacheParameterGroupFamily` - 參數群組的引擎和版本系列。例如 `memcached1.4`。
- `Description` - 使用者提供的參數群組說明。

## Example

以下範例會建立名為 myMem14 的參數群組，並使用 memcached1.4 系列做為範本。

```
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>
```

此動作的回應看起來會與以下內容相似。

```
<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>
```

建立參數群組時，它會擁有系列的預設值。若要變更預設值，您必須修改參數群組。如需更多詳細資訊，請參閱 [修改參數群組](#)。

如需更多詳細資訊，請參閱 [CreateCacheParameterGroup](#)。

## 依名稱列出參數群組

您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 來列出參數群組。

### 依名稱列出參數群組 (主控台)

下列程序說明如何使用 ElastiCache 主控台檢視參數群組清單。

### 使用 ElastiCache 主控台列出參數群組

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。

### 依名稱列出參數群組 (AWS CLI)

若要使用 AWS CLI 產生參數群組清單，請使用 `describe-cache-parameter-groups` 命令。若您提供參數群組的名稱，便只會列出該參數群組。若您沒有提供參數群組的名稱，最多會列出 `--max-records` 個參數群組。在任一種情況下，都會列出參數群組的名稱、系列和描述。

### Example

以下範本程式碼會列出 `myMem14` 參數群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-parameter-groups \  
  --cache-parameter-group-name myMem14
```

針對 Windows：

```
aws elasticache describe-cache-parameter-groups ^  
  --cache-parameter-group-name myMem14
```

此命令的輸出看起來會與以下內容相似，列出參數群組的名稱、系列和描述。

```
{  
  "CacheParameterGroups": [  
    {
```

```
    "CacheParameterGroupName": "myMem14",
    "CacheParameterGroupFamily": "memcached1.4",
    "Description": "My first parameter group"
  }
]
}
```

## Example

以下範本程式碼會列出最多 10 個參數群組。

```
aws elasticache describe-cache-parameter-groups --max-records 10
```

此命令的 JSON 輸出看起來像這樣：列出每個參數群組的名稱、系列、描述，以及在使用 redis5.6 的情況下，參數群組是否為全域資料存放區 (isGlobal) 的一部分。

```
{
  "CacheParameterGroups": [
    {
      "CacheParameterGroupName": "custom-redis32",
      "CacheParameterGroupFamily": "redis3.2",
      "Description": "custom parameter group with reserved-memory > 0"
    },
    {
      "CacheParameterGroupName": "default.memcached1.4",
      "CacheParameterGroupFamily": "memcached1.4",
      "Description": "Default parameter group for memcached1.4"
    },
    {
      "CacheParameterGroupName": "default.redis2.6",
      "CacheParameterGroupFamily": "redis2.6",
      "Description": "Default parameter group for redis2.6"
    },
    {
      "CacheParameterGroupName": "default.redis2.8",
      "CacheParameterGroupFamily": "redis2.8",
      "Description": "Default parameter group for redis2.8"
    },
    {
      "CacheParameterGroupName": "default.redis3.2",
      "CacheParameterGroupFamily": "redis3.2",
      "Description": "Default parameter group for redis3.2"
    },
  ],
}
```

```
{
  "CacheParameterGroupName": "default.redis3.2.cluster.on",
  "CacheParameterGroupFamily": "redis3.2",
  "Description": "Customized default parameter group for redis3.2 with
cluster mode on"
},
{
  "CacheParameterGroupName": "default.redis5.6.cluster.on",
  "CacheParameterGroupFamily": "redis5.0",
  "Description": "Customized default parameter group for redis5.6 with
cluster mode on",
  "isGlobal": "yes"
},
]
}
```

如需更多詳細資訊，請參閱 [describe-cache-parameter-groups](#)。

依名稱列出參數群組 (ElastiCache API)

若要使用 ElastiCache API 產生參數群組清單，請使用 DescribeCacheParameterGroups 動作。若您提供參數群組的名稱，便只會列出該參數群組。若您沒有提供參數群組的名稱，最多會列出 MaxRecords 個參數群組。在任一種情況下，都會列出參數群組的名稱、系列和描述。

### Example

以下範本程式碼會列出 myMem14 參數群組。

```
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>
```

此動作的回應看起來會與以下內容相似，列出每個參數群組的名稱、系列和描述。

```
<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

以下範本程式碼會列出最多 10 個參數群組。

```

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>

```

此動作的回應看起來像這樣：列出每個參數群組的名稱、系列、描述，以及在使用 redis5.6 的情況下，參數群組是否屬於全域資料存放區 (isGlobal)。

```

<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>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
</DescribeCacheParameterGroupsResponse>

```

```
<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>
```

如需更多詳細資訊，請參閱 [DescribeCacheParameterGroups](#)。

## 列出參數群組的值

您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 列出參數群組的參數及其值。

### 列出參數群組的值 (主控台)

下列程序說明如何使用 ElastiCache 主控台列出參數群組的參數及其值。

使用 ElastiCache 主控台列出參數群組的參數及其值

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 透過選擇參數群組名稱左側的方塊，選擇您希望列出其參數及值的參數群組。

參數及其值會在畫面底部列出。根據參數的數量，您可能需要向上或向下捲動來尋找您想要的參數。

### 列出參數群組的值 (AWS CLI)

若要使用 AWS CLI 列出參數群組的參數及其值，請使用 `describe-cache-parameters` 命令。

#### Example

以下範本程式碼會列出 `myMem14` 參數群組的所有參數及其值。

若為 Linux、macOS 或 Unix：

```
aws elasticache describe-cache-parameters \  
  --cache-parameter-group-name myMem14
```

針對 Windows：

```
aws elasticache describe-cache-parameters ^  
  --cache-parameter-group-name myMem14
```

如需更多詳細資訊，請參閱 [describe-cache-parameters](#)。

## 列出參數群組的值 (ElastiCache API)

若要使用 ElastiCache API 列出參數群組的參數及其值，請使用 DescribeCacheParameters 動作。

### Example

以下範本程式碼會列出 myMem14 參數群組的所有參數。

```
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>
```

此動作的回應看起來會與以下內容相似。此回應已進行截斷。

```
<DescribeCacheParametersResponse xmlns="http://elasticache.amazonaws.com/  
doc/2013-06-15/">  
  <DescribeCacheParametersResult>  
    <CacheClusterClassSpecificParameters>  
      <CacheNodeTypeSpecificParameter>  
        <DataType>integer</DataType>  
        <Source>system</Source>  
        <IsModifiable>>false</IsModifiable>  
        <Description>The maximum configurable amount of memory to use to store items,  
in megabytes.</Description>  
        <CacheNodeTypeSpecificValues>  
          <CacheNodeTypeSpecificValue>  
            <Value>1000</Value>  
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>  
          <CacheNodeTypeSpecificValue>  
            <Value>6000</Value>  
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>  
          <CacheNodeTypeSpecificValue>  
            <Value>7100</Value>  
            <CacheClusterClass>cache.m1.large</CacheClusterClass>  
          </CacheNodeTypeSpecificValue>
```

```
<CacheNodeTypeSpecificValue>
  <Value>1300</Value>
  <CacheClusterClass>cache.m1.small</CacheClusterClass>
</CacheNodeTypeSpecificValue>

...output omitted...

</CacheClusterClassSpecificParameters>
</DescribeCacheParametersResult>
<ResponseMetadata>
  <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
</ResponseMetadata>
</DescribeCacheParametersResponse>
```

如需更多詳細資訊，請參閱 [DescribeCacheParameters](#)。

## 修改參數群組

### Important

您無法修改任何預設參數群組。

您可以修改參數群組中的某些參數值。這些參數值都會套用到與參數群組相關聯的叢集。如需參數值變更套用到參數群組時間的詳細資訊，請參閱 [Memcached 專用參數](#)。

### 修改參數群組 (主控台)

下列程序說明如何使用 ElastiCache 主控台變更 `binding_protocol` 參數的值。您會使用相同的程序來變更任何參數的值。

### 使用 ElastiCache 主控台變更參數值

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 透過選擇參數群組名稱左側的方塊，選擇您希望修改的參數群組。  
  
參數群組的參數會在畫面底部列出。您可能需要查看各頁才能看到所有參數。
4. 若要修改一或多個參數，請選擇 Edit Parameters (編輯參數)。

5. 在 Edit Parameter Group: (編輯參數群組:) 畫面中，使用向左和向右箭頭捲動，直到找到 `binding_protocol` 參數，然後在 Value (數值) 欄中輸入 `ascii`。
6. 在 Edit Parameter Group: (編輯參數群組:) 畫面中，使用向左和向右箭頭捲動，直到找到 `cluster-enabled` 參數，然後在 Value (數值) 欄中輸入 `yes`。
7. 選擇 Save Changes (儲存變更)。
8. 若要尋找您變更的參數名稱，請參閱 [Memcached 專用參數](#)。若變更是在「重新啟動之後」才套用到參數，請重新開機每個使用此參數群組的叢集。如需詳細資訊，請參閱 [重新啟動叢集](#)。

## 修改參數群組 (AWS CLI)

若要使用 AWS CLI 變更參數值，請使用 `modify-cache-parameter-group` 命令。

### Example

若要尋找您欲變更的參數名稱及允許值，請參閱 [Memcached 專用參數](#)

下列範本程式碼會為 `myMem14` 參數群組上的兩個參數 (`chunk_size` 和 `chunk_size_growth_fact`) 設定值。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-parameter-group \  
  --cache-parameter-group-name myMem14 \  
  --parameter-name-values \  
    ParameterName=chunk_size,ParameterValue=96 \  
    ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

針對 Windows：

```
aws elasticache modify-cache-parameter-group ^  
  --cache-parameter-group-name myMem14 ^  
  --parameter-name-values ^  
    ParameterName=chunk_size,ParameterValue=96 ^  
    ParameterName=chunk_size_growth_fact,ParameterValue=1.5
```

此命令的輸出看起來會與以下內容相似。

```
{  
  "CacheParameterGroupName": "myMem14"
```

```
}
```

如需更多詳細資訊，請參閱 [modify-cache-parameter-group](#)。

### 修改參數群組 (ElastiCache API)

若要使用 ElastiCache API 變更參數群組的參數值，請使用 `ModifyCacheParameterGroup` 動作。

### Example

若要尋找您欲變更的參數名稱及允許值，請參閱 [Memcached 專用參數](#)

下列範本程式碼會為 `myMem14` 參數群組上的兩個參數 (`chunk_size` 和 `chunk_size_growth_fact`) 設定值。

```
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>
```

如需更多詳細資訊，請參閱 [ModifyCacheParameterGroup](#)。

## 刪除參數群組

您可以使用 ElastiCache 主控台、AWS CLI 或 ElastiCache API 刪除自訂參數群組。

若參數群組已和任何叢集相關聯，您便無法刪除參數群組。您也無法刪除任何預設參數群組。

### 刪除參數群組 (主控台)

下列程序說明如何使用 ElastiCache 主控台刪除參數群組。

#### 使用 ElastiCache 主控台刪除參數群組

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用參數群組的清單，請從左側的導覽窗格中，選擇 Parameter Groups (參數群組)。
3. 透過選擇參數群組名稱左側的方塊，選擇您希望刪除的參數群組。

Delete (刪除) 按鈕即變成作用中。

4. 選擇 Delete (刪除)。

Delete Parameter Group (刪除參數群組) 確認畫面隨即出現。

5. 若要刪除參數群組，請在 Delete Parameter Groups (刪除參數群組) 確認畫面上，選擇 Delete (刪除)。

若要保留參數群組，請選擇 Cancel (取消)。

### 刪除參數群組 (AWS CLI)

若要使用 AWS CLI 刪除參數群組，請使用 `delete-cache-parameter-group` 命令。針對要刪除的參數群組，以 `--cache-parameter-group-name` 指定的參數群組不能有任何與其相關聯的叢集，也不能是預設參數群組。

以下範本程式碼會刪除 `myMem14` 參數群組。

#### Example

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-cache-parameter-group \
```

```
--cache-parameter-group-name myMem14
```

針對 Windows :

```
aws elasticache delete-cache-parameter-group ^  
--cache-parameter-group-name myMem14
```

如需更多詳細資訊，請參閱 [delete-cache-parameter-group](#)。

### 刪除參數群組 (ElastiCache API)

若要使用 ElastiCache API 刪除參數群組，請使用 DeleteCacheParameterGroup 動作。針對要刪除的參數群組，以 CacheParameterGroupName 指定的參數群組不能有任何與其相關聯的叢集，也不能是預設參數群組。

### Example

以下範本程式碼會刪除 myMem14 參數群組。

```
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>
```

如需更多詳細資訊，請參閱 [DeleteCacheParameterGroup](#)。

## Memcached 專用參數

若您沒有為 Memcached 叢集指定參數群組，則會使用適合您引擎版本的預設參數群組。您無法變更預設參數群組中任何參數的值。但是，您可以建立自訂參數群組並隨時將其指派給您的叢集。如需更多詳細資訊，請參閱 [建立參數群組](#)。

### 主題

- [Memcached 1.6.17 的變更](#)
- [Memcached 1.6.6 版新增參數](#)
- [Memcached 1.5.10 參數變更](#)
- [Memcached 1.4.34 版新增參數](#)
- [Memcached 1.4.33 版新增參數](#)
- [Memcached 1.4.24 版新增參數](#)
- [Memcached 1.4.14 版新增參數](#)
- [Memcached 1.4.5 版支援參數](#)
- [Memcached 連線額外負荷](#)
- [Memcached 節點類型專用參數](#)

### Memcached 1.6.17 的變更

從 Memcached 1.6.17 開始，我們不再支援下列管理命令：`lru_crawler`、`lru` 和 `slabs`。這些變更生效後，您就無法在執行期透過命令啟用/停用 `lru_crawler`。請透過修改自訂參數群組來啟用/停用 `lru_crawler`。

### Memcached 1.6.6 版新增參數

針對 Memcached 1.6.6，沒有支援任何額外參數。

參數群組系列：`memcached1.6`

### Memcached 1.5.10 參數變更

針對 Memcached 1.5.10 版，支援以下額外參數。

參數群組系列：`memcached1.5`

名稱	詳細資訊	描述
no_modern	預設：1 類型：布林值 可修改：是 允許的值：0、1 變生效：啟動時	用於停用 slab_reassign、slab_automove、lru_crawler、lru_maintainer、maxconns_fast 命令的別名。No modern 也會將 hash_algorithm 設為 jenkins 並允許 ASCII VALUE 內嵌。適用 memcached 1.5 和更高版本。若要還原為現代，必須停用此參數並重新啟動，將會自動啟用 slab_reassign、slab_automove、lru_crawler、lru_maintainer 和 maxconns_fast。
		<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E1F5FE;"> <p> <b>Note</b></p> <p>自 2021 年 8 月 20 日起，此參數的預設組態值已從 0 變更為 1。2021 年 8 月 20 日之後，每個區域的新 ElastiCache 使用者會自動取得更新的預設值。2021 年 8 月 20 日之前區域中的現有 ElastiCache 使用者需要手動修改其自訂參數群組，才能取得這項新變更。</p> </div>
inline_ascii_resp	預設：0 類型：布林值	在項目內存放來自 VALUE 回應的數字，使用最多 24 個位元組。針對

名稱	詳細資訊	描述
	可修改：是 允許的值：0、1 變生效：啟動時	ASCII get、faster 設定小量的減速。

針對 Memcached 1.5.10，會移除以下參數。

名稱	詳細資訊	描述
expirezero_does_no_t_evict	預設：0 類型：布林值 可修改：是 允許的值：0、1 變生效：啟動時	此版本中已不再支援。
modern	預設：1 類型：布林值 可修改：是 (如果設為 no_modern，則需要重新啟動) 允許的值：0、1 變生效：啟動時	此版本中已不再支援。從此版本開始，每次啟動或重新啟動時，no-modern 預設會啟用。

### Memcached 1.4.34 版新增參數

針對 Memcached 1.4.34，沒有支援任何額外參數。

參數群組系列：memcached1.4

## Memcached 1.4.33 版新增參數

針對 Memcached 1.4.33 版，支援以下額外參數。

參數群組系列：memcached1.4

名稱	詳細資訊	描述
modern	預設：啟用 類型：布林值 可修改：是 變生效：啟動時	多種功能的別名。啟用 modern 等同於開啟以下命令，並使用 murmur3 雜湊演算法：slab_reassign、slab_auto_move、lru_crawler、lru_maintainer、maxconns_fast 及 hash_algorithm=murmur3。
watch	預設：啟用 類型：布林值 可修改：是 變生效：立即 若使用者到達其 watcher_logbuf_size 與 worker_logbuf_size 限制，則日誌可能會遭到卸除。	日誌擷取、移出或突變。例如，當使用者開啟 watch 時，他們會在 get、set、delete 或 update 發生時看到日誌。
idle_timeout	預設：0 (停用) 類型：整數 可修改：是	在詢問關閉前允許用戶端閒置的最低秒數。數值範圍：0 到 86400。

名稱	詳細資訊	描述
	變更生效：啟動時	
track_sizes	預設：停用 類型：布林值 可修改：是 變更生效：啟動時	顯示每個 Slab 群組使用的大小。  啟用 track_sizes 可讓您執行 stats sizes，而無須執行 stats sizes_enable。
watcher_logbuf_size	預設：256 (KB) 類型：整數 可修改：是 變更生效：啟動時	watch 命令會開啟 Memcached 的串流日誌。但是，watch 可能會在移出率、突變率或擷取率過高，足以導致日誌緩衝區填滿時卸除日誌。在這種情況下，使用者可增加緩衝區大小來減少遺失日誌的機率。
worker_logbuf_size	預設：64 (KB) 類型：整數 可修改：是 變更生效：啟動時	watch 命令會開啟 Memcached 的串流日誌。但是，watch 可能會在移出率、突變率或擷取率過高，足以導致日誌緩衝區填滿時卸除日誌。在這種情況下，使用者可增加緩衝區大小來減少遺失日誌的機率。
slab_chunk_max	預設：524288 (位元組) 類型：整數 可修改：是 變更生效：啟動時	指定 Slab 的大小上限。設定較小的 Slab 大小以更有效率地使用記憶體。大於 slab_chunk_max 的項目會分割成多個 Slab。

名稱	詳細資訊	描述
lru_crawler metadump [all 1 2 3]	預設：停用 類型：布林值 可修改：是 變生效：立即	若啟用 lru_crawler，則此命令會傾印所有鍵。  all 1 2 3 - 所有 Slab，或是指定特定的 Slab 編號。

## Memcached 1.4.24 版新增參數

針對 Memcached 1.4.24 版，支援以下額外參數。

參數群組系列：memcached1.4

名稱	詳細資訊	描述
disable_flush_all	預設：0 (停用) 類型：布林值 可修改：是 變生效：啟動時	新增參數 (-F) 來停用 flush_all。在您永遠不想在生產執行個體上執行完整排清時很有用。  數值：0、1 (當此值為 0 時，使用者便可執行 flush_all )。
hash_algorithm	預設：jenkins 類型：字串 可修改：是 變生效：啟動時	要使用的雜湊演算法。允許值：murmur3 及 jenkins。
lru_crawler	預設：0 (停用) 類型：布林值 可修改：是	清理已過期的項目 Slab 類別。此為在背景執行且影響程度低的程序。目前需要使用手動命令初始化網路爬取。

名稱	詳細資訊	描述
	變更生效：重新啟動後  <div data-bbox="678 373 922 800"> <p><b>Note</b></p> <p>您可以於執行時間從命令列暫時啟用 <code>lru_crawler</code>。如需詳細資訊，請參閱「描述」欄。</p> </div>	若要暫時啟用，請在命令列執行 <code>lru_crawler enable</code> 。  <code>lru_crawler 1,3,5</code> 會網路爬取 Slab 類別 1、3 及 5，尋找過期的項目，並新增到釋放清單。  數值：0、1  <div data-bbox="1036 625 1479 1010"> <p><b>Note</b></p> <p>在命令列啟用 <code>lru_crawler</code> 會啟用網路爬取程式，直到在命令列停用或下一次重新開機為止。若要永久啟用，您必須修改參數值。如需更多詳細資訊，請參閱 <a href="#">修改參數群組</a>。</p> </div>
<code>lru_maintainer</code>	預設：0 (停用)  類型：布林值  可修改：是  變更生效：啟動時	背景執行緒，可在到達容量時於 LRU 間隨機移動項目。數值：0、1。

名稱	詳細資訊	描述
expirezero_does_no_t_evict	預設：0 (停用) 類型：布林值 可修改：是 變生效：啟動時	搭配 lru_maintainer 使用時，讓過期時間為 0 的項目無法移出。 <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;">  <b>Warning</b>              這可能會填滿其他可移出項目的可用記憶體。           </div> 可設為略過 lru_maintainer 。

### Memcached 1.4.14 版新增參數

針對 Memcached 1.4.14 版，支援以下額外參數。

參數群組系列：memcached1.4

### Memcached 1.4.14 版中新增的參數

名稱	描述
config_max	ElastiCache 組態項目的數量上限。

名稱	描述
config_size_max	組態項目的大小上限，單位為位元組。
hashpower_init	ElastiCache 雜湊表的初始大小，以 2 的乘冪表示。預設為 16 ( $2^{16}$ )，或 65536 個鍵。

名稱	描述
maxconns_fast	<p>變更到達連線數量上限時處理新連線請求的方式。若將此參數設為 0 (零)，則新的連線會新增到待處理項目佇列，並等待其他連線關閉。若將此參數設為 1，ElastiCache 會將錯誤傳送至用戶端，並立即關閉連線。</p>

名稱	描述
slab_automove	<p>調整 Slab 自動移動演算法：若將此參數設為 0 (零)，則停用自動移動演算法。若設為 1，ElastiCache 會採取緩慢、保守的方式來自動移動 Slab。若設為 2，ElastiCache 會在發生移出時積極移動 Slab。(除非用於測試用途，否則不建議此模式。)</p>

名稱	描述
slab_reassign	啟用或停用 Slab 指派。若將此參數設為 1，您可以使用 "slabs reassign" 命令來手動重新指派記憶體。

## Memcached 1.4.5 版支援參數

參數群組系列：memcached1.4

針對 Memcached 1.4.5 版，支援以下參數。

## Memcached 1.4.5 版中新增的參數

名稱	詳細資訊	描述
backlog_queue_limit	預設：1024 類型：整數 可修改：否	待處理佇列限制。
binding_protocol	預設：自動 類型：字串 可修改：是 變更生效：重新啟動後	繫結通訊協定。 允許值：ascii 及 auto。 如需修改 binding_protocol 值的指導，請參閱 <a href="#">修改參數群組</a> 。
cas_disabled	預設值：0 (false) 類型：布林值 可修改：是 變更生效：重新啟動後	若為 1 (True)，則會停用檢查及設定 (CAS) 操作，並且存放的項目會比啟用 CAS 時少使用 8 個位元組。
chunk_size	預設：48 類型：整數 可修改：是 變更生效：重新啟動後	配置給最小項目鍵、值及標記的空間數量下限 (位元組)。
chunk_size_growth_factor	預設：1.25 類型：浮動 可修改：是 變更生效：重新啟動後	成長因數，控制每個後續 Memcached 區塊的大小。每個區塊都會比前一個區塊大 chunk_size_growth_factor 倍。

名稱	詳細資訊	描述
error_on_memory_exhausted	預設值：0 (false) 類型：布林值 可修改：是 變生效：重新啟動後	若為 1 (True)，則當沒有記憶體可存放項目時，Memcached 會傳回錯誤而非移出項目。
large_memory_pages	預設值：0 (false) 類型：布林值 可修改：否	若為 1 (True)，ElastiCache 會嘗試使用大型記憶體分頁。
lock_down_paged_memory	預設值：0 (false) 類型：布林值 可修改：否	若為 1 (True)，ElastiCache 會鎖定所有分頁記憶體。
max_item_size	預設：1048576 類型：整數 可修改：是 變生效：重新啟動後	可存放在叢集中的最大項目大小 (位元組)。
max_simultaneous_connections	預設：65000 類型：整數 可修改：否	同時連線數量上限。
maximize_core_file_limit	預設值：0 (false) 類型：布林值 可修改： 變生效：重新啟動後	若為 1 (True)，ElastiCache 會最大化核心檔案限制。

名稱	詳細資訊	描述
memcached_connections_overhead	預設：100 類型：整數 可修改：是 變生效：重新啟動後	要為 Memcached 連線及其他雜項額外負荷預留的記憶體數量。如需此參數的相關資訊，請參閱 <a href="#">Memcached 連線額外負荷</a> 。
requests_per_event	預設：20 類型：整數 可修改：否	指定連線每個事件的請求數上限。此限制為避免資源耗盡的必要項目。

## Memcached 連線額外負荷

在每個節點上，可供存放項目的記憶體為該節點可用記憶體的總數 (該值存放在 `max_cache_memory` 參數中)，減去用於連線及其他額外負荷 (該值存放在 `memcached_connections_overhead` 參數中) 所使用的記憶體。例如，`cache.m1.small` 類型的節點具有 1300MB 的 `max_cache_memory`。使用預設 100MB 的 `memcached_connections_overhead` 值，Memcached 程序便會擁有 1200MB，可用於存放項目。

`memcached_connections_overhead` 參數的預設值足以滿足大多數的使用案例。但是，連線額外負荷所需要的配置數量可能會因多項因素而不同，包括請求率、承載大小，以及連線數量。

您可以變更 `memcached_connections_overhead` 的值來更進一步配合您應用程式的需求。例如，增加 `memcached_connections_overhead` 參數的值會減少可用於存放項目的記憶體數量，為連線額外負荷提供更大的緩衝區。減少 `memcached_connections_overhead` 參數的值可讓您擁有更多記憶體來存放項目，但可能會增加切換 (swap) 用量及降低效能。若您察覺切換 (swap) 用量及效能降低，請嘗試增加 `memcached_connections_overhead` 參數的值。

### Important

針對 `cache.t1.micro` 節點類型，`memcached_connections_overhead` 的值會透過以下方式決定：

- 若您的叢集使用預設參數群組，ElastiCache 會將 `memcached_connections_overhead` 的值設為 13MB。

- 若您的叢集使用您自行建立的參數群組，您可以將 `memcached_connections_overhead` 的值設為您所選擇的值。

## Memcached 節點類型專用參數

雖然大多數的參數都只有單一值，有些參數則可能會根據所使用的節點類型而有不同的值。下表顯示每個節點類型 `max_cache_memory` 及 `num_threads` 參數的預設值。這些參數上的值都無法修改。

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.t1.micro	213	1
cache.t2.micro	555	1
cache.t2.small	1588	1
cache.t2.medium	3301	2
cache.t3.micro	512	2
cache.t3.small	1402	2
cache.t3.medium	3364	2
cache.t4g.micro	512	2
cache.t4g.small	1402	2
cache.t4g.medium	3164	2
cache.m1.small	1301	1
cache.m1.medium	3350	1
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	33800	2

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.m2.2xlarge	30412	4
cache.m2.4xlarge	68000	16
cache.m3.medium	2850	1
cache.m3.large	6200	2
cache.m3.xlarge	13600	4
cache.m3.2xlarge	28600	8
cache.m4.large	6573	2
cache.m4.xlarge	11496	4
cache.m4.2xlarge	30412	8
cache.m4.4xlarge	62234	16
cache.m4.10xlarge	158355	40
cache.m5.large	6537	2
cache.m5.xlarge	13248	4
cache.m5.2xlarge	26671	8
cache.m5.4xlarge	53516	16
cache.m5.12xlarge	160900	48
cache.m5.24xlarge	321865	96
cache.m6g.large	6537	2
cache.m6g.xlarge	13248	4
cache.m6g.2xlarge	26671	8

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.m6g.4xlarge	53516	16
cache.m6g.8xlarge	107000	32
cache.m6g.12xlarge	160900	48
cache.m6g.16xlarge	214577	64
cache.c1.xlarge	6600	8
cache.r3.large	13800	2
cache.r3.xlarge	29100	4
cache.r3.2xlarge	59600	8
cache.r3.4xlarge	120600	16
cache.r3.8xlarge	120600	32
cache.r4.large	12590	2
cache.r4.xlarge	25652	4
cache.r4.2xlarge	51686	8
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

節點類型	max_cache_memory (以 MB 為單位)	num_threads
cache.r5.12xlarge	325400	48
cache.r5.24xlarge	650869	96
cache.r6g.large	13387	2
cache.r6g.xlarge	26953	4
cache.r6g.2xlarge	54084	8
cache.r6g.4xlarge	108347	16
cache.r6g.8xlarge	214577	32
cache.r6g.12xlarge	325400	48
cache.r6g.16xlarge	429154	64
cache.c7gn.large	3164	2
cache.c7gn.xlarge	6537	4
cache.c7gn.2xlarge	13248	8
cache.c7gn.4xlarge	26671	16
cache.c7gn.8xlarge	53516	32
cache.c7gn.12xlarge	325400	48
cache.c7gn.16xlarge	108347	64

 Note

所有 T2 執行個體都是在 Amazon Virtual Private Cloud (Amazon VPC) 中建立的。

# 縮放記憶 ElastiCache 體快取

## 縮放記憶 ElastiCache 體快取

ElastiCache 無伺服器會在工作負載流量上升或下降時自動容納您的工作負載流量。針對每個 ElastiCache 無伺服器快取，ElastiCache 持續追蹤資源 (例如 CPU、記憶體和網路) 的使用率。當任何這些資源受到限制時，ElastiCache Serverless 會透過新增碎片並將資料重新分配到新的碎片來向外擴充，而不會對應用程式造成任何停機時間。您可以監視快取資料儲存體和 ElastiCacheProcessingUnits (ECPU) 計算使用BytesUsedForCache量的測量結果，CloudWatch 來監視快取使用的資源。

## 設定擴展限制以管理成本

您可以選擇為快取設定快取資料儲存和 ECPU/秒的最大用量，以控制快取成本。這樣做將確保您的快取用量永遠不會超過所設定的上限。

如果您設定了擴展上限，當快取達到上限時，應用程式可能會遇到快取效能降低的情況。當您設定快取資料儲存最大值且快取資料儲存達到最大值時，ElastiCache 將開始使用 LRU 邏輯收回快取中的資料。當您設定 ECPU/秒上限且工作負載的運算使用率超過此值時，ElastiCache 將開始限制 Memcached 要求。

如果您在BytesUsedForCache或上設定最大限制ElastiCacheProcessingUnits，我們強烈建議您將CloudWatch 警示設定為低於最大限制的值，以便在快取運作接近這些限制時收到通知。建議設定的警示值為上限的 75%。請參閱有關如何設定 CloudWatch 鬧鐘的文件。

## 使用無伺服器預先調整 ElastiCache

### ElastiCache 無伺服器預先調整

透過預先調整 (也稱為預熱)，您可以為快取設定支援的 ElastiCache 最低限制。您可以為每秒 ElastiCache 處理單位 (ECPU) 或資料儲存設定這些最小值。這對於準備預期的擴展事件很有用。例如，如果一家遊戲公司預計在新遊戲推出的第一分鐘內登入會增加 5 倍，他們就可以準備好快取，以應付這種顯著的使用量激增。

您可以使用 ElastiCache 主控台、CLI 或 API 執行預先調整規模。ElastiCache 無伺服器會在 60 分鐘內更新快取記憶體上的可用 ECP，並在最小限制更新完成時傳送事件通知。

### 預縮放的運作方式

當 ECPUS/ 秒或資料儲存體的最低限制透過主控台、CLI 或 API 更新時，該新限制將在 1 小時內提供。ElastiCache 無伺服器在空白快取上支援每秒 30K ECPUS，使用「從複本讀取」功能時，最高可支援每秒 90K ECPUS。ElastiCache 每 10-12 分鐘可以加倍/秒。這種擴展速度足以滿足大多數工作負載。如果您預期即將到來的擴展事件可能會超過此速率，那麼我們建議您將最小 ECPUS /秒設定為期望在高峰事件發生前至少 60 分鐘的峰值 ECPUS /秒。否則，應用程式可能會遇到更高的延遲和請求限制。

最低限制更新完成後，ElastiCache 無伺服器會開始計量新的每秒最低 ECPU 或新的最低儲存空間。即使您的應用程式未在快取上執行要求，或您的資料儲存使用量低於最低限度，也會發生這種情況。當您從目前的設定降低最小限制時，會立即更新，因此 ElastiCache 無伺服器會立即開始以新的最低限制計量。

#### Note

- 當您設定最低用量限制時，即使您的實際使用量低於最低用量限制，仍需支付該限制的費用。超過最低使用量限制的 ECPU 或資料儲存使用量將按一般費率計費。例如，如果您將最低使用量限制設定為 100,000 個 ECP/秒，則即使您的使用量低於設定的最低使用量，仍需支付每小時至少 1.224 美元的費用（使用 us-east-1 中的 ECPU 價格）。
- ElastiCache 無伺服器支援快取上彙總層級的要求最小規模。ElastiCache 無伺服器也支援每個插槽最多每秒 30K ECPUS (使用唯讀連線使用「從複本讀取」時，每秒為 90K ECPUS)。最佳做法是，您的應用程式應確保 Redis 插槽間的金鑰散佈和金鑰間的流量盡可能一致。

## 使用主控台和設定縮放限制 AWS CLI

### 使用 AWS 主控台設定擴展限制

1. 請登入 AWS Management Console 並開啟 ElastiCache 主控台，[網址為 https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/)。
2. 在導覽窗格中，選擇您要修改之快取上執行的引擎。
3. 接著會出現執行所選引擎的快取清單。
4. 選擇快取名稱左側的選項按鈕來選擇要修改的快取。
5. 選擇 Actions (動作)，然後選擇 Modify (修改)。
6. 在使用量限制下，設定適當的記憶體或運算限制。
7. 按一下預覽變更，然後儲存變更。

## 使用設定縮放限制 AWS CLI

若要使用 CLI 變更擴展限制，請使用 `modify-serverless-cache` API。

Linux：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

Windows：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^  
--cache-usage-limits 'DataStorage={Minimum=10,Maximum=100,Unit=GB},  
ECPUPerSecond={Minimum=1000,Maximum=100000}'
```

## 使用 CLI 移除擴展限制

若要使用 CLI 移除縮放限制，請將「最小」和「最大限制」參數設定為 0。

Linux：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> \  
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},  
ECPUPerSecond={Minimum=0,Maximum=0}'
```

Windows：

```
aws elasticache modify-serverless-cache --serverless-cache-name <cache name> ^  
--cache-usage-limits 'DataStorage={Minimum=0,Maximum=0,Unit=GB},  
ECPUPerSecond={Minimum=0,Maximum=0}'
```

## ElastiCache 為記憶體快取自行設計的叢集擴充

您的應用程式需要處理的資料量通常是動態的。它會隨著您的業務成長或遇到需求的一般波動而增加或減少。如果您自行管理快取，您必須為需求尖峰佈建足夠的硬體，這可能所費不貲。透過使用 Amazon，ElastiCache 您可以擴展以滿足目前的需求，只需按使用量付費。ElastiCache 可讓您擴充快取以符合需求。

下列各項能幫助您找到用於您要執行之擴展動作的正確主題。

## 擴展 Memcached 叢集

動作	主題
向外擴展	<a href="#">將節點新增至叢集</a>
向內擴展	<a href="#">從叢集刪除節點</a>
變更節點類型	<a href="#">垂直擴展 Memcached</a>

記憶體快取的叢集由 1 到 60 個節點組成。向外和向內擴展 Memcached 叢集，就如同從叢集新增或移除節點般容易。

如果您在 Memcached 叢集中需要 60 個以上的節點，或是一個 AWS 區域中的節點總計超過 300 個，請在 <https://aws.amazon.com/contact-us/elasticache-node-limit-request/> 填寫提高 ElastiCache 限制要求表單。

因為您可以在 Memcached 叢集的所有節點間分割資料，您很少需要向上擴展至具有更大記憶體的節點類型。不過，因為 Memcached 引擎不會持續保存資料，如果您確實擴展至不同的節點類型，您的新叢集一開始會是空白的，除非您的應用程式填入資料。

### 主題

- [水平擴展 Memcached](#)
- [垂直擴展 Memcached](#)

## 水平擴展 Memcached

Memcached 引擎支援在多個節點間分割您的資料。因此，Memcached 叢集可輕鬆水平擴展。記憶體快取的叢集可以有 1 到 60 個節點。若要水平擴展 Memcached 叢集，只需要新增或移除節點。

如果您在 Memcached 叢集中需要 60 個以上的節點，或是一個 AWS 區域中的節點總計超過 300 個，請在 <https://aws.amazon.com/contact-us/elasticache-node-limit-request/> 填寫提高 ElastiCache 限制要求表單。

下列主題詳細說明如何透過新增或移除節點，向外或向內擴展 Memcached 叢集。

- [將節點新增至叢集](#)
- [從您的叢集刪除節點](#)

每次在 Memcached 叢集中變更節點的數量，您必須至少為您的一些金鑰空間進行重新映射，使得它可映射到正確的節點。如需有關負載平衡 Memcached 叢集的詳細資訊，請參閱[設定您的 ElastiCache 用戶端以達到高效率的負載平衡](#)。

如果是在 Memcached 叢集上使用自動搜索，則不需在您新增或移除節點時，變更應用程式中的端點。如需自動探索的詳細資訊，請參閱[自動識別叢集中的節點](#)。如果您不使用自動搜索，每次在 Memcached 叢集中變更節點的數量時，即必須更新應用程式中的端點。

## 垂直擴展 Memcached

向上或向下擴展 Memcached 叢集時，您必須建立新的叢集。Memcached 叢集一開始一律是空白的，除非您的應用程式填入資料。

### Important

如果您要向下擴展至較小的節點類型，請確定該較小的節點類型對您的資料和成本而言已足夠。如需詳細資訊，請參閱[選取快取節點大小](#)。

## 主題

- [垂直擴展 Memcached \(主控台\)](#)
- [垂直擴展 Memcached \(AWS CLI\)](#)
- [垂直縮放內存緩存 \(ElastiCache API\)](#)

## 垂直擴展 Memcached (主控台)

下列程序將逐步引導您使用 ElastiCache 主控台垂直擴展叢集。

### 垂直擴展 Memcached 叢集 (主控台)

1. 使用新的節點類型來建立新叢集。如需詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。
2. 在應用程式中，將端點更新為新叢集的端點。如需詳細資訊，請參閱[尋找叢集的端點 \(主控台\)](#)。
3. 刪除舊叢集。如需詳細資訊，請參閱[刪除 Memcached 中的新節點](#)。

## 垂直擴展 Memcached (AWS CLI)

下列程序會逐步引導您使用 AWS CLI，來垂直擴展 Memcached 快取叢集。

## 垂直擴展 Memcached 快取叢集 (AWS CLI)

1. 使用新的節點類型來建立新快取叢集。如需詳細資訊，請參閱[使用 CLI 建立叢集](#)。
2. 在應用程式中，將端點更新為新叢集的端點。如需詳細資訊，請參閱[尋找端點 \(AWS CLI\)](#)。
3. 刪除舊快取叢集。如需詳細資訊，請參閱[使用 AWS CLI](#)。

## 垂直縮放內存緩存 (ElastiCache API)

下列程序會引導您逐步使用 ElastiCache API 來調整 Memcached 快取叢集的垂直規模。

### 若要垂直縮放記憶體快取叢集 (ElastiCache API)

1. 使用新的節點類型來建立新快取叢集。如需更多資訊，請參閱[建立叢集 \(ElastiCache API\)](#)
2. 在應用程式中，將端點更新為新快取叢集的端點。如需詳細資訊，請參閱[尋找端點 \(ElastiCache API\)](#)。
3. 刪除舊快取叢集。如需更多詳細資訊，請參閱[使用 ElastiCache API](#)。

# 為您的 ElastiCache 資源加上標籤

為協助您管理叢集和其他 ElastiCache 資源，您可以用標籤形式將自己的中繼資料指派給每個資源。標籤可讓您以不同的方式分類您的 AWS 資源，例如依據目的、擁有者或環境。當您有許多相同類型的資源時，這將會很有用，因為—您可以依據先前指派的標籤，快速識別特定的資源。本主題說明標籤並示範如何建立它們。

## Warning

根據最佳實務，建議您不要在標籤中包含敏感資料。

## 標籤基本概念

標籤是您指派給 AWS 資源的標籤。每個標籤皆包含由您定義的一個金鑰與一個選用值。標籤可讓您以不同的方式分類您的 AWS 資源，例如依據用途或擁有者。例如，您可以為您帳戶的 ElastiCache 叢集定義一組標籤，協助您追蹤各個執行個體的擁有者與使用者群組。

我們建議您為每種資源類型建立符合您需求的標籤金鑰。使用一致的標籤金鑰組可讓您更輕鬆管理您的資源。您可以根據您新增的標籤搜尋和篩選資源。如需如何實作有效資源標記策略的詳細資訊，請參閱 [AWS 白皮書標記最佳實務](#)。

標籤對 ElastiCache 來說不具任何語意意義，且會嚴格解譯為字元字串。此外，標籤不會自動指派給您的資源。您可以編輯標籤金鑰和值，並且可以隨時從資源移除標籤。您可以將標籤的值設為 null。若您將與現有標籤具有相同鍵的標籤新增到該資源，則新值會覆寫舊值。如果您刪除資源，也會刪除任何該資源的標籤。此外，如果您為複寫群組新增或刪除標籤，也會為該複寫群組中的所有節點新增或移除其標籤。

您可以透過 AWS Management Console、AWS CLI 和 ElastiCache API 使用標籤。

若使用 IAM，您可以控制 AWS 帳戶中的哪些使用者具有建立、編輯和刪除標籤的許可。如需更多詳細資訊，請參閱 [資源層級許可](#)。

## 您可以標記的資源

您可以為帳戶中現有的大多數 ElastiCache 資源新增標籤。下表列出支援標籤建立的資源。若您使用 AWS Management Console，可以利用 [標籤編輯器](#) 來對資源套用標籤。有些資源畫面可讓您在建立資源時指定資源的標籤；例如，具有 Name 索引鍵和您指定值的標籤。在大多數的案例中，主控台會立

即在建立資源後套用標籤 (而非在資源建立過程時)。主控台可根據 Name 標籤整理資源，但此標籤對 ElastiCache 服務來說不具任何語意意義。

此外，有些資源建立動作可讓您在建立資源時指定資源的標籤。若標籤無法在資源建立時套用，我們會轉返資源建立程序。這可確保資源不是具有標籤建立，就是不會建立，因此無論何時都不會有不具有標籤的資源。藉由在建立時為資源建立標籤，您可以消除在資源建立後執行自訂標籤指令碼的必要。

如果使用 Amazon ElastiCache API、AWS CLI 或 AWS 軟體開發套件，則可以對相關 ElastiCache API 動作使用 Tags 參數，將標籤套用到新資源。這些類別為：

- CreateServerlessCache
- CreateCacheCluster
- CreateCacheParameterGroup
- CreateCacheSecurityGroup
- CreateCacheSubnetGroup
- PurchaseReservedCacheNodesOffering

下表說明可加上標籤的 ElastiCache 資源，以及可在使用 ElastiCache API、AWS CLI 或 AWS 軟體開發套件建立時加上標籤的資源。

#### ElastiCache 資源的標記支援

支援標籤	支援在建立時標記
是	是
是	是
是	是
是	是

支援標籤	支援在建立時標記
是	是
是	是

您可以在 IAM 政策中將標籤式的資源層級許可套用到支援在建立時新增標籤的 ElastiCache API 動作，以對可在建立時為資源加上標籤的使用者和群組實作精密控制。您的資源從建立時便已獲得妥善的保護，標籤會立即套用到您的資源。因此，控制資源使用情況的任何標籤式資源層級許可都會立即生效。您可以更準確的追蹤和報告您的資源。您可以強制新資源使用標籤，並控制哪些標籤金鑰和值會在您的資源上設定。

如需更多詳細資訊，請參閱 [為資源加上標籤的範例](#)。

如需為資源加上標籤以便計費的詳細資訊，請參閱「[使用成本配置標籤監控成本](#)」。

## 標籤限制

以下基本限制適用於標籤：

- 每一資源最多標籤數 - 50
- 對於每一個資源，每個標籤金鑰必須是唯一的，且每個標籤金鑰只能有一個值。
- 索引鍵長度上限 - 128 個 UTF-8 Unicode 字元。
- 值長度上限 - 256 個 UTF-8 Unicode 字元。
- 雖然 ElastiCache 允許在標籤中使用任何字元，但其他服務的限制可能更嚴格。服務間允許的字元包括：可用 UTF-8 表示的英文字母、數字和空格，還有以下字元：+ - = . \_ : / @
- 標籤金鑰與值皆區分大小寫。
- 此 aws: 字首已保留供 AWS 使用。如果標籤具有此字首的標籤金鑰，則您無法編輯或刪除標籤的金鑰或值。具 aws: 字首的標籤，不算在受資源限制的標籤計數內。

您無法僅根據標籤終止、停止或刪除資源。您必須指定資源識別符。例如，若要刪除您套用稱為 DeleteMe 標籤金鑰的快照，您必須搭配快照的資源識別符 (例如 DeleteSnapshot) 使用 snap-1234567890abcdef0 動作。

如需深入了解可加上標籤的 ElastiCache 資源，請參閱「[您可以標記的資源](#)」。

## 為資源加上標籤的範例

- 使用標籤建立無伺服器快取

```
aws elasticache create-serverless-cache \  
  --serverless-cache-name CacheName \  
  --engine memcached \  
  --tags Key="Cost Center", Value="1110001" Key="project",Value="XYZ"
```

- 新增標籤至無伺服器快取

```
aws elasticache add-tags-to-resource \  
  --resource-name arn:aws:elasticache:us-east-1:111111222233:serverlesscache:my-cache \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

- 使用標籤建立快取叢集。

```
aws elasticache create-cache-cluster \  
  --cluster-id testing-tags \  
  --cluster-description cluster-test \  
  --cache-subnet-group-name test \  
  --cache-node-type cache.t2.micro \  
  --engine memcached \  
  --tags Key="project",Value="XYZ" Key="Elasticache",Value="Service"
```

## 標籤型存取控制政策範例

1. 只有在叢集具有 Project=XYZ 標籤時，才允許 AddTagsToResource 動作傳送至叢集。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "elasticache:AddTagsToResource",  
      "Resource": [  
        "arn:aws:elasticache:*:*:cluster:*"  
      ],  
    },  
  ],  
}
```

```

        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/Project": "XYZ"
            }
        }
    ]
}

```

2. 如果複寫群組包含 Project 和 Service 標籤，且索引鍵與 Project 和 Service 不同，便允許該複寫群組的 RemoveTagsFromResource 動作。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticache:RemoveTagsFromResource",
      "Resource": [
        "arn:aws:elasticache:*:*:replicationgroup:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Service": "Elasticache",
          "aws:ResourceTag/Project": "XYZ"
        },
        "ForAnyValue:StringNotEqualsIgnoreCase": {
          "aws:TagKeys": [
            "Project",
            "Service"
          ]
        }
      }
    }
  ]
}

```

3. 只有在標籤不是 Project 和 Service 時，才允許對任何資源執行 AddTagsToResource。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

    "Effect": "Allow",
    "Action": "elasticache:AddTagsToResource",
    "Resource": [
        "arn:aws:elasticache:*:*:*:*"
    ],
    "Condition": {
        "ForAnyValue:StringNotEqualsIgnoreCase": {
            "aws:TagKeys": [
                "Service",
                "Project"
            ]
        }
    }
}
]
}

```

4. 如果請求標籤 CreateCacheCluster 遺失或不等於 Project、Dev 或 QA，則拒絕 Prod 動作。

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*",
                "arn:aws:elasticache:*:*:securitygroup:*",
                "arn:aws:elasticache:*:*:replicationgroup:*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:cluster:*"
            ],
            "Condition": {
                "Null": {

```

```
        "aws:RequestTag/Project": "true"
    }
}
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:AddTagsToResource"
    ],
    "Resource": "arn:aws:elasticache:*:*:cluster:*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Project": [
                "Dev",
                "Prod",
                "QA"
            ]
        }
    }
}
]
```

如需條件索引鍵的相關資訊，請參閱 [使用條件索引鍵](#)。

## 使用成本配置標籤監控成本

將成本配置標籤新增至 Amazon ElastiCache 中的資源時，您可以將發票上的費用依資源標籤值分組，藉此追蹤成本。

ElastiCache 成本配置標籤是您所定義並與 ElastiCache 資源建立關聯的索引鍵值組。索引鍵與值皆會區分大小寫。您可以使用標籤索引鍵來定義類別，而標籤值可為該類別中的某個項目。例如，您可以定義標籤索引鍵 `CostCenter`，且標籤值為 `10010`，指出資源是指派給 `10010` 成本中心。您也可利用 `Environment` 之類的索引鍵，和 `test` 或 `production` 之類的值，以使用標籤來指定用於測試或生產的資源。我們建議您使用一組一致的標籤索引鍵，讓您更輕鬆地追蹤與資源關聯的成本。

您可以使用成本配置標籤來整理您的 AWS 帳單，以反映您自己的成本結構。方式是註冊以取得包含標籤鍵值的 AWS 帳戶帳單。接著，若要查看合併資源的成本，請根據具有相同標籤鍵值的資源來整理您的帳單資訊。例如，您可以使用特定應用程式名稱來標記數個資源，然後整理帳單資訊以查看該應用程式跨數項服務的總成本。

您也可以結合標籤來以更高的細節層次追蹤成本。例如，若要按區域追蹤您的服務成本，您可以使用標籤索引鍵 `Service` 和 `Region`。在某個資源上，您會有值 `ElastiCache` 和 `Asia Pacific (Singapore)`，而在另一個資源上，則有值 `ElastiCache` 和 `Europe (Frankfurt)`。如此一來您就可以依區域劃分來查看您的 ElastiCache 總成本。如需詳細資訊，請參閱《AWS Billing 使用者指南》中的[使用成本分配標籤](#)。

您可以將 ElastiCache 成本配置標籤新增至 Memcached 叢集。新增、列出、修改、複製或移除標籤時，該操作只會套用至指定的叢集。

### ElastiCache 成本配置標籤的特性

- 成本配置標籤會套用至 ElastiCache 資源，這些資源是在 CLI 和 API 作業中以 ARN 的形式指定。資源類型會是「叢集」。

ARN 範例：`arn:aws:elasticache:<region>:<customer-id>:<resource-type>:<resource-name>`

Memcached：標籤僅會套用至叢集。

範例 ARN：`arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

- 標籤金鑰是標籤必要的名稱。索引鍵的字串值的長度可以是 1 到 128 個 Unicode 字元，不可在前面加上 `aws:`。此字串只能包含一組 Unicode 字母、數字、空格、底線 (`_`)、句點 (`.`)、冒號 (`:`)、反斜線 (`\`)、等號 (`=`)、加號 (`+`)、連字號 (`-`) 或 `@` 符號 (`@`)。
- 標籤值為標籤的選用值。值的字串值長度可以是 1 到 256 個 Unicode 字元，不可在前面加上 `aws:`。此字串只能包含一組 Unicode 字母、數字、空格、底線 (`_`)、句點 (`.`)、冒號 (`:`)、反斜線 (`\`)、等號 (`=`)、加號 (`+`)、連字號 (`-`) 或 `@` 符號 (`@`)。
- ElastiCache 資源最多可有 50 個標籤。
- 標籤組中的值不必是唯一的。例如，您可以有一個標籤組，其中的索引鍵 `Service` 和 `Application` 都有值 `ElastiCache`。

AWS 不會將任何語意意義套用至標籤。標籤將嚴格解譯為字元字串。AWS 不會對任何 ElastiCache 資源自動設定任何標籤。

## 使用 AWS CLI 管理成本配置標籤

您可以使用 AWS CLI 來新增、修改或移除成本配置標籤。

成本配置標籤會套用至 ElastiCache for Memcached 叢集。要加標籤的叢集是使用 ARN (Amazon Resource Name) 來指定。

範例 ARN : `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

範例 ARN : `arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster`

### 主題

- [使用 AWS CLI 列出標籤](#)
- [使用 AWS CLI 新增標籤](#)
- [使用 AWS CLI 修改標籤](#)
- [使用 AWS CLI 移除標籤](#)

## 使用 AWS CLI 列出標籤

您可以使用 AWS CLI，透過 [list-tags-for-resource](#) 作業列出現有 ElastiCache 資源上的標籤。

下列程式碼使用 AWS CLI 來列出 us-west-2 區域中 Memcached 叢集 my-cluster 上的標籤。

若為 Linux、macOS 或 Unix：

```
aws elasticache list-tags-for-resource \  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

針對 Windows：

```
aws elasticache list-tags-for-resource ^  
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster
```

此操作的輸出看起來應該類似以下，這是資源上所有標籤的清單。

```
{  
  "TagList": [  
    {  
      "Value": "10110",
```

```
    "Key": "CostCenter"
  },
  {
    "Value": "EC2",
    "Key": "Service"
  }
]
```

如果資源上沒有標籤，輸出會是空的標籤清單。

```
{
  "TagList": []
}
```

如需詳細資訊，請參閱 ElastiCache 的 AWS CLI 「[list-tags-for-resource](#)」。

## 使用 AWS CLI 新增標籤

您可以使用 AWS CLI，透過 [add-tags-to-resource](#) CLI 作業將標籤新增至現有的 ElastiCache 資源。如果標籤索引鍵不存在於資源上，則索引鍵和值會新增至資源。如果索引鍵已存在於資源上，則與該索引鍵相關聯的值會更新為新的值。

下列程式碼使用 AWS CLI 將索引鍵 Service 和 Region (值分別為 elasticache 和 us-west-2) 新增至 us-west-2 區域中叢集 my-cluster-001 裡的。

若為 Linux、macOS 或 Unix：

```
aws elasticache add-tags-to-resource \
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \
  --tags Key=Service,Value=elasticache \
         Key=Region,Value=us-west-2
```

針對 Windows：

```
aws elasticache add-tags-to-resource ^
  --resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^
  --tags Key=Service,Value=elasticache ^
         Key=Region,Value=us-west-2
```

此操作的輸出看起來應該類似以下，這是在操作後資源上所有標籤的清單。

```
{
  "TagList": [
    {
      "Value": "elasticache",
      "Key": "Service"
    },
    {
      "Value": "us-west-2",
      "Key": "Region"
    }
  ]
}
```

如需詳細資訊，請參閱 ElastiCache 的 AWS CLI 「[add-tags-to-resource](#)」。

您也可以使用 AWS CLI 在建立新叢集時將標籤新增至叢集，方法是使用操作 [create-cache-cluster](#)。使用 ElastiCache 管理主控台建立叢集時無法新增標籤。建立叢集之後，您可以接著使用主控台來將標籤新增至叢集。

## 使用 AWS CLI 修改標籤

您可以使用 AWS CLI 來修改 ElastiCache for Memcached 叢集上的標籤。

若要修改標籤：

- 使用 [add-tags-to-resource](#) 來新增標籤和值，或是變更與現有標籤關聯的值。
- 使用 [remove-tags-from-resource](#) 來從資源移除指定的標籤。

這兩項操作的輸出會是指定叢集上標籤和其值的清單。

## 使用 AWS CLI 移除標籤

您可以使用 AWS CLI，透過 [remove-tags-from-resource](#) 作業從現有 ElastiCache for Memcached 叢集移除標籤。

下列程式碼使用 AWS CLI 將附有索引鍵 Service 和 Region 的標籤從 us-west-2 區域中叢集 my-cluster-001 裡的中移除。

若為 Linux、macOS 或 Unix：

```
aws elasticache remove-tags-from-resource \
```

```
--resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster \  
--tag-keys PM Service
```

針對 Windows :

```
aws elasticache remove-tags-from-resource ^  
--resource-name arn:aws:elasticache:us-west-2:0123456789:cluster:my-cluster ^  
--tag-keys PM Service
```

此操作的輸出看起來應該類似以下，這是在操作後資源上所有標籤的清單。

```
{  
  "TagList": []  
}
```

如需詳細資訊，請參閱 ElastiCache 的 AWS CLI 「[remove-tags-from-resource](#)」。

## 使用 ElastiCache API 管理成本配置標籤

您可以使用 ElastiCache API 來新增、修改或移除成本配置標籤。

成本配置標籤會套用至 ElastiCache for Memcached 叢集。要加標籤的叢集是使用 ARN (Amazon Resource Name) 來指定。

範例 ARN : arn:aws:elasticache:us-west-2:1234567890:cluster:my-cluster

主題

- [使用 ElastiCache API 列出標籤](#)
- [使用 ElastiCache API 新增標籤](#)
- [使用 ElastiCache API 修改標籤](#)
- [使用 ElastiCache API 移除標籤](#)

## 使用 ElastiCache API 列出標籤

您可以使用 ElastiCache API，透過 [ListTagsForResource](#) 作業列出現有資源上的標籤。

下列程式碼使用 ElastiCache API 來列出 us-west-2 區域中 my-cluster 資源的標籤。

```
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>
```

## 使用 ElastiCache API 新增標籤

您可以使用 ElastiCache API，透過 [AddTagsToResource](#) 作業將標籤新增至現有的 ElastiCache 叢集。如果標籤索引鍵不存在於資源上，則索引鍵和值會新增至資源。如果索引鍵已存在於資源上，則與該索引鍵相關聯的值會更新為新的值。

下列程式碼使用 ElastiCache API，將索引鍵 Region 和 elasticache (值分別為 Service 和 us-west-2) 新增至 us-west-2 區域中的 my-cluster 資源。

```
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>
```

如需詳細資訊，請參閱 Amazon ElastiCache API 參考中的「[AddTagsToResource](#)」。

## 使用 ElastiCache API 修改標籤

您可以使用 ElastiCache API 來修改 ElastiCache 叢集上的標籤。

若要修改標籤的值：

- 使用 [AddTagsToResource](#) 操作來新增標籤和值，或是變更現有標籤的值。
- 使用 [RemoveTagsFromResource](#) 來從資源移除標籤。

這兩項操作的輸出會是指定資源上標籤和其值的清單。

使用 [RemoveTagsFromResource](#) 來從資源移除標籤。

## 使用 ElastiCache API 移除標籤

您可以使用 ElastiCache API，透過 [RemoveTagsFromResource](#) 作業從現有 ElastiCache for Memcached 叢集移除標籤。

下列程式碼使用 ElastiCache API，將含有索引鍵 Service 和 Region 的標籤從 us-west-2 區域中 my-cluster 叢集裡的 移除。

```
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>
```

## 使用 Amazon ElastiCache Well-Architected Lens

本節描述 Amazon ElastiCache Well-Architected Lens，它提供了一套設計原則和指引，有助於設計出架構良好的 ElastiCache 工作負載。

- ElastiCache Lens 可加入 [AWS Well-Architected Framework](#) 之中。
- 每個支柱都有一組問題，可幫助您展開有關 ElastiCache 架構的討論。
  - 每個問題都有一些引導練習，且配有分數以便製作報告。
    - 必要 - 進入 prod 之前的必要前提 (缺少代表高風險)
    - 最佳 - 客戶能達到的最佳狀態
    - 良好 - 我們建議客戶達到的狀態 (缺少代表中等風險)
- Well-Architected 的術語
  - [元件](#)：程式碼、組態及 AWS 資源，這些會一起交付以滿足需求。元件會與其他元件互動，而且經常等同於微型服務架構中的服務。
  - [工作負載](#)：共同提供商業價值的一組元件。工作負載的範例包括行銷網站、電子商務網站、行動應用程式後端系統、分析平台等。

## 主題

- [Amazon ElastiCache Well-Architected Lens 卓越運作支柱](#)
- [Amazon ElastiCache Well-Architected Lens 安全支柱](#)
- [Amazon ElastiCache Well-Architected Lens 可靠性支柱](#)
- [Amazon ElastiCache Well-Architected Lens 效能效率支柱](#)
- [Amazon ElastiCache Well-Architected Lens 成本最佳化支柱](#)

## Amazon ElastiCache Well-Architected Lens 卓越運作支柱

卓越營運支柱著重於執行和監控系統，以提供商業價值並持續改善流程和程序。重要主題包括自動化變更、回應事件，以及定義管理日常作業的標準。

### 主題

- [OE 1：如何了解及回應 ElastiCache 叢集觸發的警示和事件？](#)
- [OE 2：何時及如何擴展您現有的 ElastiCache 叢集？](#)
- [OE 3：如何管理 ElastiCache 叢集資源並讓叢集保持最新狀態？](#)
- [OE 4：如何管理用戶端與您的 ElastiCache 叢集的連線？](#)
- [OE 5：如何為工作負載部署 ElastiCache 元件？](#)
- [OE 6：如何規劃故障因應措施及減少故障？](#)
- [OE 7：如何對 Redis 引擎事件進行故障診斷？](#)

### OE 1：如何了解及回應 ElastiCache 叢集觸發的警示和事件？

**問題難易度簡介：**當您操作 ElastiCache 叢集時，您可以選擇在特定事件發生時接收通知和警示。根據預設，ElastiCache 會記錄與資源相關的事件，例如容錯移轉、節點取代、擴展操作、排程維護等。每個事件都包含日期與時間、來源名稱與來源類型，以及說明。

**問題難易度優點：**只要能夠了解並管理觸發叢集產生警示的事件背後的基本原因，就能更有效地操作並適當回應事件。

- [必要] 在 ElastiCache 主控台上 (選取您的區域之後) 或使用 [Amazon 命令列界面](#) (AWS CLI) [describe-events](#) 命令和 [ElastiCache API](#)，檢閱 ElastiCache 產生的事件。將 ElastiCache 設定為使用 Amazon Simple Notification Service (Amazon SNS) 傳送重要叢集事件的通知。使用 Amazon SNS 搭配您的叢集，就可透過程式設計的方式對 ElastiCache 事件採取行動。

- 事件分成兩大類：目前事件和排定事件。目前事件清單包括：資源建立和刪除、擴展操作、容錯移轉、節點重新開機、建立快照、叢集參數修改、CA 憑證更新、失敗事件 (叢集佈建失敗 - VPC 或 ENI、擴展失敗 - ENI 和快照失敗)。排定事件清單包括：排定在維護時段進行更換的節點，以及重新排定的節點更換作業。
- 雖然您不需要立即回應其中一些事件，但務必先查看所有失敗事件：
  - ElastiCache:AddCacheNodeFailed
  - ElastiCache:CacheClusterProvisioningFailed
  - ElastiCache:CacheClusterScalingFailed
  - ElastiCache:CacheNodesRebooted
  - ElastiCache:SnapshotFailed (僅 Redis)
- [資源]：
  - [管理 ElastiCache Amazon SNS 通知](#)
  - [事件通知和 Amazon SNS](#)
- [最佳] 若要自動回應事件，請利用 AWS 產品和服務功能，如 SNS 和 Lambda 函數。遵循最佳實務進行小量、頻繁、可恢復的變更，以程式碼形式隨著時間讓您的操作演進。您應使用 Amazon CloudWatch 指標來監控您的叢集。

[資源]：針對使用 Lambda 和 SNS 的使用案例，[使用 AWS Lambda、Amazon Route 53 和 Amazon SNS 監控 Amazon ElastiCache for Redis \(叢集模式已停用\) 讀取複本端點](#)。

## OE 2：何時及如何擴展您現有的 ElastiCache 叢集？

問題難易度簡介：將 ElastiCache 叢集調整成適當規模是一種平衡動作，只要基礎工作負載類型發生變更就需評估該動作。您的目標是讓您的工作負載在適當規模的環境下運作。

問題難易度優點：資源過度利用可能會導致延遲增加且整體效能降低。另一方面來說，使用不足可能會導致資源過度佈建，而以非最佳成本最佳化。只要適當調整環境的規模，您就能在效能效率與成本最佳化之間取得平衡。為了補救資源過度利用或利用不足的情形，ElastiCache 可採取兩種維度進行縮減。您可以增加或減少節點容量來垂直擴展。您也可以新增和移除節點來水平擴展。

- [必要] 解決主節點上 CPU 和網路過度利用的方法，是卸載讀取操作並將其重新導向至複本節點。使用複本節點進行讀取操作，可降低主節點的使用率。您可在 Redis 用戶端程式庫中進行這項設定，方法是連線至 ElastiCache 讀取器端點 (若叢集模式已停用)，或使用 Redis READONLY 命令 (若叢集模式已啟用)。

[資源]：

- [尋找連線端點](#)
- [適當調整叢集規模](#)
- [Redis READONLY 命令](#)
- [必要] 監控重要叢集資源 (如 CPU、記憶體和網路) 的使用率。需追蹤這些特定叢集資源的使用率，才能在擴展和擴展操作類型方面做出明智的決策。ElastiCache for Redis (叢集模式已停用) 的主節點和複本節點都可垂直擴展。複本節點也可以從 0 到 5 個節點水平擴展。若叢集模式已啟用，叢集的每個碎片內情況也相同。此外，您可以增加或減少碎片的數量。

[資源]：

- [使用 Amazon CloudWatch 搭配 Amazon ElastiCache for Redis 進行監控的最佳實務](#)
- [擴展 ElastiCache for Redis 叢集](#)
- [擴展 ElastiCache for Memcached 叢集](#)
- [最佳] 隨著時間監控趨勢可幫助您偵測到在特定時間點監控時，可能無法察覺的工作負載變更。若要偵測更長期的趨勢，請使用 CloudWatch 指標來掃描更長的時間範圍。長期觀測 CloudWatch 指標所獲得的經驗，應能幫助您預測叢集資源使用率。CloudWatch 資料點和指標最多可使用 455 天。

[資源]：

- [使用 CloudWatch 指標監控 ElastiCache for Redis](#)
- [使用 CloudWatch 指標監控 Memcached](#)
- [使用 Amazon CloudWatch 搭配 Amazon ElastiCache for Redis 進行監控的最佳實務](#)
- [最佳] 如果您的 ElastiCache 資源是使用 CloudFormation 所建立，則最佳實務是使用 CloudFormation 範本進行變更，以保持操作一致性，並避免發生未受管理的組態變更和堆疊漂移。

[資源]：

- [CloudFormation 的 ElastiCache 資源類型參考](#)
- [最佳] 使用叢集操作資料自動化擴展操作，並在 CloudWatch 中定義閾值以設定警示。使用 CloudWatch 事件和 Simple Notification Service (SNS) 觸發 Lambda 函數，並執行 ElastiCache API 以自動擴展您的叢集。例如，在 EngineCPUUtilization 指標經過長時間之後達到 80% 時，將碎片新增至叢集。另一個選項是使用 DatabaseMemoryUsedPercentages 作為記憶體型閾值。

[資源]：

- [使用 Amazon CloudWatch 警示](#)
- [什麼是 Amazon CloudWatch 事件？](#)
- [使用 AWS Lambda 搭配 Amazon Simple Notification Service](#)

- [ElastiCache API 參考](#)

## OE 3：如何管理 ElastiCache 叢集資源並讓叢集保持最新狀態？

**問題難易度簡介：**大規模操作時，您必須能夠精確找出並識別所有 ElastiCache 資源。推出新的應用程式功能時，您需要在所有 ElastiCache 環境類型中建立對稱的叢集版本：開發、測試和生產。資源屬性可讓您針對不同的操作目標分隔環境，例如在推出新功能和啟用新的安全機制時。

**問題難易度優點：**將開發、測試和生產環境加以分隔，是最佳操作實務。在整個環境中利用充分了解並妥善記載的程序對叢集和節點套用最新的軟體修補程式，同樣也是最佳實務。利用原生 ElastiCache 功能可讓您的工程團隊專注於達成業務目標，而不需分心處理 ElastiCache 維護工作。

- [最佳] 在可用的最新引擎版本上執行，並且一推出自助式更新就盡快套用。ElastiCache 會在您指定的叢集維護時段自動更新其基礎設施。不過，叢集中執行的節點則會透過自助式更新進行更新。這些更新有兩種類型：安全修補程式或次要軟體更新。您務必了解修補程式類型的差異，以及套用的時機。

[資源]：

- [Amazon ElastiCache 中的自助式更新](#)
- [Amazon ElastiCache 受管維護與服務更新說明頁面](#)
- [最佳] 使用標籤來整理您的 ElastiCache 資源。在複寫群組上使用標籤，而非在個別節點上使用。您可以設定在查詢資源時顯示的標籤，也可以使用標籤來執行搜尋和套用篩選器。您可使用資源群組輕鬆建立和維護擁有共同標籤集的資源集合。

[資源]：

- [標記最佳實務](#)
- [CloudFormation 的 ElastiCache 資源類型參考](#)
- [參數群組](#)

## OE 4：如何管理用戶端與您的 ElastiCache 叢集的連線？

**問題難易度簡介：**大規模操作時，您需要了解用戶端如何與 ElastiCache 叢集連線，以便管理應用程式操作層面 (例如回應時間)。

**問題難易度優點：**選擇最合適的連線機制，可確保應用程式不會因為連線錯誤 (例如逾時) 而中斷連線。

- [必要] 將讀取與寫入操作分開，並連線至複本節點來執行讀取操作。但請注意，當您將寫入與讀取分開時，由於 Redis 複寫為非同步性質，因此您將無法在寫入金鑰之後立即讀取該金鑰。WAIT 命令可用來改善真實世界的資料安全性，並強制複本先確認寫入才能回應用戶端，但要付出整體效能成本的代價。您可以在 ElastiCache for Redis 用戶端程式庫中，使用 ElastiCache 讀取器端點 (若叢集模式已停用) 來設定使用複本節點進行讀取操作。若叢集模式已啟用，請使用 ElastiCache for Redis READONLY 命令。對於許多 ElastiCache for Redis 用戶端程式庫來說，ElastiCache for Redis READONLY 命令是預設實作，或透過組態設定實作。

[資源]：

- [尋找連線端點](#)
- [READONLY](#)

- [必要] 使用連線集區。無論在用戶端或伺服器端建立 TCP 連線，都會伴隨 CPU 時間成本，而集區可讓您重複使用 TCP 連線。

為了減少連線額外負荷，建議您使用連線集區。有了連線集區，您的應用程式就可以「隨意」重複使用和釋出連線，而不會因建立連線而產生成本。您可以透過 ElastiCache for Redis 用戶端程式庫 (如支援) 實作連線集區，並提供適用您應用程式環境的架構，或是從頭建立新的架構。

- [最佳] 確實將用戶端的通訊端逾時設定為至少 1 秒 (相較於數個用戶端中的一般預設值「無」)。
  - 若設定的逾時值太低，可能導致伺服器負載較高時發生逾時。若設定的值太高，則可能導致您的應用程式花費很長的時間來偵測連線問題。
  - 藉由在用戶端應用程式中實作連線集區來控制新連線的數量。這樣做可降低開啟和關閉連線所需的延遲和 CPU 使用率，並且在叢集上已啟用 TLS 的情況下執行 TLS 交握。

[資源]：[設定 Amazon ElastiCache for Redis 以提高可用性](#)

- [良好] 使用管道傳輸 (您的使用案例允許的話) 可大幅提高效能。
  - 使用管道傳輸可減少應用程式用戶端與叢集之間的往返時間 (RTT)，而且即使用戶端尚未讀取先前的回應，也可以處理新的請求。
  - 使用管道傳輸可一次將多個命令傳送至伺服器，而不需等待回覆/確認。管道傳輸的缺點在於，當您最後大量截取所有回應時，可能已有錯誤發生，但您直到最後才察覺到。
  - 當傳回錯誤而忽略錯誤請求時，實作方法來重試請求。

[資源]：[管道傳輸](#)

## OE 5：如何為工作負載部署 ElastiCache 元件？

**問題難易度簡介：** ElastiCache 環境可透過 AWS 主控台手動部署，或透過 API、CLI、工具組等以程式設計方式部署。卓越運作最佳實務建議您，盡可能透過程式碼自動化部署。此外，ElastiCache 叢集可依工作負載加以區隔，也可相互結合以達到成本最佳化。

**問題難易度優點：** 為您的 ElastiCache 環境選擇最合適的部署機制，就能隨著時間推移提升卓越運作的成效。建議您盡可能以程式碼形式執行操作，以便盡量減少人為錯誤並增加事件的重複能力、彈性和回應時間。

只要了解工作負載隔離的需求，您就可以選擇讓每一項工作負載擁有專用的 ElastiCache 環境，或將多個工作負載結合為單一叢集或其組合。了解當中的權衡，有助於在卓越運作和成本最佳化之間取得平衡

- [必要] 了解 ElastiCache 可用的部署選項，並盡可能將這些程序自動化。可能的自動化途徑包括 CloudFormation、AWS CLI/SDK 及 API。

[資源]：

- [Amazon ElastiCache 資源類型參考](#)
- [elasticache](#)
- [Amazon ElastiCache API 參考](#)
- [必要] 為所有工作負載決定所需的叢集隔離層級。
  - [最佳]：高度隔離 - 工作負載對叢集的對應為 1:1。能夠以每個工作負載為基礎，對 ElastiCache 資源的存取、調整大小、擴展及管理實施最精細的控制。
  - [較佳]：中度隔離 - 依用途採 M:1 隔離，但可能跨多個工作負載共用 (例如，一個叢集專門用來快取工作負載，而另一個叢集專門用來傳訊)。
  - [良好]：低度隔離 - 所有用途均採 M:1 隔離，完全共用。建議用於可接受共用存取的工作負載。

## OE 6：如何規劃故障因應措施及減少故障？

**問題難易度簡介：** 卓越運作包括執行定期的「預先檢測」活動來預測故障情形，以找出可能的故障來源，有利於及早將其移除或緩解。ElastiCache 提供虛容錯移轉 API，可基於測試目的用於模擬節點故障事件。

**問題難易度優點：** 透過提前測試故障情況，您就可以了解它們如何影響您的工作負載。這樣就能安全測試回應程序及其有效性，並且讓您的團隊熟悉其執行過程。

[必要] 定期以開發/測試帳戶執行容錯移轉測試。 [TestFailover](#)

## OE 7：如何對 Redis 引擎事件進行故障診斷？

**問題難易度簡介：**卓越運作需要能夠同時調查服務層級和引擎層級的資訊，以分析叢集的運作狀況和狀態。Amazon ElastiCache for Redis 可同時對 Amazon CloudWatch 和 Amazon Kinesis Data Firehose 發送 Redis 引擎日誌。

**問題難易度優點：**在 Amazon ElastiCache for Redis 叢集上啟用 Redis 引擎日誌，就能針對影響叢集運作狀態和效能的事件提供深入洞察。Redis 引擎日誌會直接從 Redis 引擎提供資料，這些資料無法透過 ElastiCache 事件機制取得。只要仔細觀測 ElastiCache 事件 (請參閱前段的 OE-1) 和 Redis 引擎日誌，就能在從 ElastiCache 服務的觀點和 Redis 引擎的觀點進行故障診斷時，判斷出事件的順序。

- [必要] 確定已啟用 Redis 引擎記錄功能，此功能自 ElastiCache for Redis 6.2 及更新版本開始提供。此功能可在建立叢集期間啟用，或是在建立後，藉由修改叢集來啟用。
- 判斷 Amazon CloudWatch Logs 或 Amazon Kinesis Data Firehose 適合作為 Redis 引擎日誌的目標。
- 在 CloudWatch 或 Kinesis Data Firehose 內選取適當的目標日誌，以用來保留日誌。如果您有多個叢集，請考慮讓每個叢集使用不同的目標日誌，因為這樣做有助於在故障診斷時隔離資料。

[資源]：

- 日誌傳送：[日誌傳送](#)
- 記錄目的地：[Amazon CloudWatch Logs](#)
- Amazon CloudWatch Logs 簡介：[什麼是 Amazon CloudWatch Logs？](#)
- Amazon Kinesis Data Firehose 簡介：[什麼是 Amazon Kinesis Data Firehose？](#)
- [最佳] 如果使用 Amazon CloudWatch Logs，請考慮利用 Amazon CloudWatch Logs Insights 來查詢 Redis 引擎日誌以取得重要資訊。

舉例來說，針對 CloudWatch 日誌群組建立查詢，該群組包含 Redis 引擎日誌，而日誌中會傳回 LogLevel 為「WARNING」(警告)的事件，例如：

```
fields @timestamp, LogLevel, Message
| sort @timestamp desc
| filter LogLevel = "WARNING"
```

[資源]：[使用 CloudWatch Logs Insights 分析日誌資料](#)

## Amazon ElastiCache Well-Architected Lens 安全支柱

安全支柱著重於保護資訊和系統。重要主題包括資料的機密性和完整性、識別和管理誰可以透過權限為主的管理做什麼、保護系統，以及建立控制項來偵測安全事件。

### 主題

- [SEC 1：您採取哪些步驟來控制授權存取 ElastiCache 資料的行為？](#)
- [SEC 2：您的應用程式是否需有額外的授權才能對 ElastiCache 進行網路為主及更高的控制？](#)
- [SEC 3：是否存在可能不小心執行命令而造成資料遺失或故障的風險？](#)
- [SEC 4：如何使用 ElastiCache 確保靜態資料加密](#)
- [SEC 5：如何使用 ElastiCache 加密傳輸中的資料？](#)
- [SEC 6：如何限制對控制平面資源的存取權？](#)
- [SEC 7：如何偵測和回應安全事件？](#)

### SEC 1：您採取哪些步驟來控制授權存取 ElastiCache 資料的行為？

問題難易度簡介：所有 ElastiCache 叢集都設計為可從 VPC 中的 Amazon Elastic Compute Cloud 執行個體、無伺服器函數 (AWS Lambda) 或容器 (Amazon Elastic Container Service) 進行存取。最常遇到的情況是從同一個 Amazon Virtual Private Cloud (Amazon Virtual Private Cloud) 內的 Amazon Elastic Compute Cloud 執行個體存取 ElastiCache 叢集。您必須先授權讓 Amazon EC2 執行個體存取叢集，才能從 Amazon EC2 執行個體連線至叢集。若要存取 VPC 中執行的 ElastiCache 叢集，則必須對叢集的網路輸入權限。

問題難易度優點：進入叢集的網路輸入是透過 VPC 安全性群組控制。安全群組會做為您 Amazon EC2 執行個體的虛擬防火牆，負責控制傳入及傳出流量。傳入規則會控制傳入至您的執行個體的流量，以及傳出規則會控制從您的執行個體傳出的流量。以 ElastiCache 為例，啟動叢集時，它需要與安全群組建立關聯。這樣可確保組成叢集的所有節點都已設有傳入和傳出流量規則。此外，ElastiCache 會設定為只能在私有子網路上部署，如此就只能透過 VPC 的私有網路存取這些子網路。

- [必要] 與叢集相關聯的安全群組會控制叢集的網路輸入和存取。根據預設，安全群組不會定義任何傳入規則，因此沒有 ElastiCache 的輸入路徑。若要啟用此項，請在安全群組上設定傳入規則以指定來源 IP 地址/範圍、TCP 類型流量，以及 ElastiCache 叢集的連接埠 (例如 ElastiCache for Redis 使用的預設連接埠 6379)。雖然可以允許一組非常廣泛的輸入來源，例如 VPC 內的所有資源 (0.0.0.0/0)，但建議盡可能精確定義傳入規則，例如，對於在與特定安全群組相關聯的 Amazon EC2 執行個體上執行的 Redis 用戶端，僅授權傳入存取權。

[資源]：

- [子網路和子網路群組](#)
- [存取您的叢集或複寫群組](#)
- [使用安全群組控制前往資源的流量](#)
- [適用於 Linux 執行個體的 Amazon Elastic Compute Cloud 安全群組](#)
- [必要] 可將 AWS Identity and Access Management 政策指派給 AWS Lambda 函數，使其能夠存取 ElastiCache 資料。若要啟用此功能，請建立具有 AWSLambdaVPCLambdaAccessExecutionRole 許可的 IAM 執行角色，然後將角色指派給 AWS Lambda 函數。

[資源]：設定 Lambda 函數使其能夠存取 Amazon VPC 中的 Amazon ElastiCache：[教學課程：設定 Lambda 函數使其能夠存取 Amazon VPC 中的 Amazon ElastiCache](#)

## SEC 2：您的應用程式是否需有額外的授權才能對 ElastiCache 進行網路為主及更高的控制？

問題難易度簡介：在必須於個別用戶端層級限制或控制 ElastiCache for Redis 叢集存取權的情況下，建議您透過 ElastiCache for Redis AUTH 命令進行驗證。ElastiCache for Redis 身分驗證字符搭配選用的使用者和使用者群組管理功能時，可讓 ElastiCache for Redis 在允許用戶端執行命令和存取金鑰之前要求輸入密碼，藉此提高資料平面的安全性。

問題難易度優點：為了協助確保資料安全，ElastiCache for Redis 提供了幾種機制來防範未經授權存取資料的行為。當中包括強制執行角色型存取控制 (RBAC) AUTH，或是強制用戶端使用 AUTH 字符 (密碼) 來連線到 ElastiCache，然後才能執行授權的命令。

- [最佳]針對 ElastiCache for Redis 6.x 或更高版本，藉由定義使用者群組、使用者及存取字串來定義身分驗證和授權控制項。將使用者指派至使用者群組，然後將使用者群組指派至叢集。若要使用 RBAC，則必須在建立叢集時選取它，並且必須啟用傳輸中加密。確定您使用的 Redis 用戶端可支援 TLS，這樣才能使用 RBAC。

[資源]：

- [將 RBAC 套用至 ElastiCache for Redis 的複寫群組](#)
- [使用存取字串指定許可](#)
- [ACL](#)
- [支援的 ElastiCache for Redis 版本](#)
- [最佳] 針對 ElastiCache for Redis 6.x 之前的版本，除了針對 ElastiCache for Redis AUTH 設定強式字符/密碼並維護嚴格的密碼政策之外，最佳實務就是輪替密碼/字符。ElastiCache 在任一特定時間點最多可管理兩 (2) 個身分驗證字符。您也可以修改叢集，以明確要求使用身分驗證字符。

[資源] : [修改現有 ElastiCache for Redis 叢集上的 AUTH 字符](#)

### SEC 3 : 是否存在可能不小心執行命令而造成資料遺失或故障的風險？

問題難易度簡介：有一些 Redis 命令在錯誤執行或遭到惡意行為者執行的情況下，可能會對操作產生不利的影響。從效能和資料安全的角度來看，這些命令可能會產生預料之外的後果。例如，開發人員可能會在開發環境中例行呼叫 FLUSHALL 命令，而由於錯誤可能會在生產系統上不小心嘗試呼叫此命令，進而導致意外的資料遺失。

問題難易度優點：自 ElastiCache for Redis 5.0.3 開始，您能夠在 ElastiCache 上重新命名某些可能對工作負載造成干擾的命令。重新命名命令有助於防止它們在叢集上意外執行。

• [必要]

[資源] :

- [ElastiCache for Redis 5.0.3 版 \(已移除，使用 5.0.6 版\)](#)
- [Redis 5.0.3 參數變更](#)
- [Redis 安全](#)

### SEC 4 : 如何使用 ElastiCache 確保靜態資料加密

問題難易度簡介：雖然 ElastiCache for Redis 是記憶體內資料存放區，但可以加密任何可能在叢集的標準操作過程中保存 (儲存) 的資料。這包括寫入 Amazon S3 的排程和手動備份，以及因同步和交換操作而儲存到磁碟儲存空間的資料。M6g 和 R6g 系列中的執行個體類型也具有永遠開啟記憶體內加密功能。

問題難易度優點：ElastiCache for Redis 提供了選用的靜態加密功能，以提高資料安全性。

• [必要] 只有在建立 ElastiCache 叢集 (複寫群組) 後，才能啟用靜態加密。無法藉由修改現有叢集來開始加密靜態資料。根據預設，ElastiCache 會提供和管理靜態加密中使用的金鑰。

[資源] :

- [靜態加密條件](#)
  - [啟用靜態加密](#)
- [最佳] 利用 Amazon EC2 執行個體類型，它會在資料位於記憶體中時加密資料 (例如 M6g 或 R6g)。盡可能考慮管理自己的靜態加密金鑰。對於更嚴格的資料安全環境，可使用 AWS Key Management

Service (KMS) 自行管理客戶主金鑰 (CMK)。透過整合 ElastiCache 與 AWS Key Management Service，您就能建立、擁有及管理用於加密 ElastiCache for Redis 叢集的靜態資料的金鑰。

[資源]：

- [使用 AWS Key Management Service 中的客戶自管金鑰](#)
- [AWS Key Management Service](#)
- [AWS KMS 概念](#)

## SEC 5：如何使用 ElastiCache 加密傳輸中的資料？

問題難易度簡介：一般常會要求在傳輸過程中減少資料遭到入侵的情況。這代表分散式系統元件內的資料，以及應用程式用戶端和叢集節點之間的資料。ElastiCache for Redis 可支援這項需求，因為它允許加密用戶端和叢集之間的傳輸中資料，以及叢集節點彼此之間的傳輸中資料。M6g 和 R6g 系列中的執行個體類型也具有永遠開啟記憶體內加密功能。

問題難易度優點：Amazon ElastiCache 傳輸中加密是一項選用功能，可讓您在資料最易遭受攻擊的點，也就是資料在地點間傳輸時，增加資料的安全性。

- [必要] 只有在建立 ElastiCache for Redis 叢集 (複寫群組) 之時才能啟用傳輸中加密。請注意，由於加密/解密資料需要進行額外處理，因此實作傳輸中加密會對效能造成一些影響。若要了解其影響，建議您分別在啟用傳輸中加密之前和之後建立工作負載的基準。

[資源]：

- [傳輸中加密概觀](#)

## SEC 6：如何限制對控制平面資源的存取權？

問題難易度簡介：IAM 政策和 ARN 可針對 ElastiCache for Redis 進行精細的存取控制，進而更嚴格地控制建立、修改和刪除 ElastiCache for Redis 叢集的管理工作。

問題難易度優點：Amazon ElastiCache 資源 (例如複寫群組、節點等) 的管理工作可根據 IAM 政策限於具有特定許可的 AWS 帳戶，藉此改善資源的安全和可靠性。

- [必要] 藉由指派特定 AWS Identity and Access Management 政策給 AWS 使用者來管理 Amazon ElastiCache 資源的存取權，就可更精細地控制哪些帳戶能夠在叢集上執行哪些動作。

[資源]：

- [管理您的 ElastiCache 資源的存取許可概觀](#)

- [針對 Amazon ElastiCache 使用以身分為基礎的政策 \(IAM 政策\)](#)

## SEC 7：如何偵測和回應安全事件？

問題難易度簡介：在啟用 RBAC 的情況下進行部署時，ElastiCache 會匯出 CloudWatch 指標以通知使用者有安全事件。這些指標有助於找出企圖進行驗證、存取金鑰，或連線的 RBAC 使用者企圖執行未獲授權之命令的失敗嘗試。

此外，AWS 產品和服務資源可藉由自動化部署及記錄所有動作與修改以供日後檢閱/稽核，以協助保護整體工作負載。

問題難易度優點：藉由監控事件可讓您的組織根據您的需求、政策和程序做出回應。自動監控和回應這些安全事件，可強化您的整體安全態勢。

- [必要] 熟悉與 RBAC 身分驗證和授權失敗相關的已發佈 CloudWatch 指標。
  - AuthenticationFailures = 嘗試對 Redis 進行身分驗證失敗
  - KeyAuthorizationFailures = 使用者嘗試在未經許可的情況下存取金鑰失敗
  - CommandAuthorizationFailures = 使用者嘗試在未經許可的情況下執行命令失敗

[資源]：

- [Redis 的指標](#)
- [最佳] 建議在這些指標上設定警示和通知，並視需要做出回應。

[資源]：

- [使用 Amazon CloudWatch 警示](#)
- [最佳] 使用 Redis ACL LOG 命令收集進一步的詳細資訊

[資源]：

- [ACL LOG](#)
- [最佳] 熟悉 AWS 產品和服務功能，因為它與監控、記錄及分析 ElastiCache 部署和事件相關

[資源]：

- [使用 AWS CloudTrail 記錄 Amazon ElastiCache API 呼叫](#)
- [elasticache-redis-cluster-automatic-backup-check](#)
- [使用 CloudWatch 指標監控用量](#)

# Amazon ElastiCache Well-Architected Lens 可靠性支柱

## 主題

- [REL 1：如何支援高可用性 \(HA\) 架構部署？](#)
- [REL 2：如何使用 ElastiCache 來達成您的復原點目標 \(RPO\)？](#)
- [REL 3：如何支援災難復原 \(DR\) 需求？](#)
- [REL 4：如何有效地規劃容錯移轉？](#)
- [REL 5：您的 ElastiCache 元件是否設計為可擴展？](#)

## REL 1：如何支援高可用性 (HA) 架構部署？

**問題難易度簡介：**了解 Amazon ElastiCache 的高可用性架構，將可讓您在可用性事件期間彈性地進行操作。

**問題難易度優點：**架構 ElastiCache 叢集使其能夠彈性地從故障中恢復，就可確保提高 ElastiCache 部署的可用性。

- [必要] 判斷您的 ElastiCache 叢集需要的可靠性層級。不同的工作負載具有不同的彈性標準，從完全暫時性工作負載到關鍵任務工作負載都有。為您操作所在的每一種環境 (例如開發、測試和生產) 定義需求。

**快取引擎：**比較 Memcached 與 ElastiCache for Redis

1. Memcached 不提供任何複寫機制，主要用於暫時性工作負載。
  2. ElastiCache for Redis 提供 HA 功能，下面將深入討論這些功能
- [最佳] 對於需要 HA 的工作負載，使用叢集模式的 ElastiCache for Redis，且每個碎片至少有兩個複本，即使對於只需要一個碎片這類輸送量需求不高的工作負載也是如此。
    1. 若叢集模式已啟用，則會自動啟用多可用區。

多可用區可在進行任何規劃或意外的維護工作以及減少可用區域故障時。藉由自動從主節點容錯移轉至複本的方式，將停機時間降至最低。

2. 對於碎片化的工作負載，至少有三個碎片可在容錯移轉事件期間協助加速復原，因為 Redis 叢集通訊協定要求大多數的主節點為可用狀態以達到規定數量。
3. 在整體可用性中設定兩個或多個複本。

若有兩個複本，就能在其中一個複本進行維護的情況下，改善讀取可擴展性以及讀取可用性。

4. 使用以 Graviton2 為基礎的節點類型 (大多數區域中的預設節點)。

Amazon ElastiCache for Redis 已為這些節點增添了最佳化的效能。因此，您可以獲得更好的複寫和同步處理效能，進而改善整體可用性。

5. 監控並調整適當大小以應付預期的流量尖峰：在高負載的情況下，ElastiCache for Redis 引擎可能會無回應，進而影響可用性。BytesUsedForCache 和 DatabaseMemoryUsagePercentage 是記憶體用量的良好指標，而 ReplicationLag 是您根據寫入速率判斷複寫運作狀態的指標。您可以使用這些指標來觸發叢集擴展。
6. 利用 [在生產容錯移轉事件之前容錯移轉 API](#) 進行測試，確保用戶端的彈性。

[資源]：

- [設定 Amazon ElastiCache for Redis 以提高可用性](#)
- [使用複寫群組的高可用性](#)

## REL 2：如何使用 ElastiCache 來達成您的復原點目標 (RPO)？

問題難易度簡介：了解工作負載 RPO，以做出明智的 ElastiCache 備份和復原策略決策。

問題難易度優點：備妥 RPO 策略就可在發生災難復原的情況下，改善業務連續性。設計備份與還原政策可協助您達成 ElastiCache 資料的復原點目標 (RPO)。ElastiCache for Redis 提供了存放在 Amazon S3 中的快照功能，以及可設定的保留政策。這些快照會在定義的備份時段拍攝，並由服務自動處理。如果您的工作負載需要更精細程度的備份，您可以選擇每天最多建立 20 個手動備份。手動建立的備份不受服務保留政策的約束，可以無限期保留。

- [必要] 了解並記錄 ElastiCache 部署的 RPO。
  - 請注意，Memcached 不提供任何備份程序。
  - 檢閱 ElastiCache 備份與還原功能的各項功能。
- [最佳] 備妥通訊良好的程序來備份叢集。
  - 視需要啟動手動備份。
  - 檢閱自動備份的保留政策。
  - 請注意，手動備份將無限期保留。
  - 將自動備份排程在低使用量的期間進行。
  - 針對讀取複本執行備份操作，以確保對叢集效能的影響降到最低。
- [良好] 利用排程的 ElastiCache 備份功能，在既定時段定期備份您的資料。
  - 定期測試從備份還原的程序。
- [資源]：

- [Redis](#)
- [ElastiCache for Redis 的備份與還原](#)
- [建立手動備份](#)
- [排程自動備份](#)
- [備份與還原 ElastiCache Redis 叢集](#)

## REL 3：如何支援災難復原 (DR) 需求？

問題難易度簡介：災難復原是任何工作負載規劃的重要環節。ElastiCache for Redis 提供了多種選項，可根據工作負載彈性需求實作災難復原。只要使用 Amazon ElastiCache for Redis 全域資料存放區，就可以寫入您在某一個區域中的 ElastiCache for Redis 叢集，並且讓資料能夠從另兩個跨區域複本叢集讀取，藉此實現跨區域的低延遲讀取和災難復原。

問題難易度優點：了解各種災難情境並規劃因應措施，就可確保業務連續性。災難復原策略必須在成本、效能影響和可能的資料遺失之間取得平衡。

- [必要] 根據工作負載需求，為您所有的 ElastiCache 元件制定並記錄災難復原策略。ElastiCache 的獨特之處在於，有些使用案例是完全暫時性的，不需要任何災難復原策略，而有些使用案例則完全相反，需要極為健全的災難復原策略。所有選項都必須針對成本最佳化加以權衡，也就是說，彈性越大，所需的基礎設施數量也越多。

了解區域層級和多區域層級可用的災難復原選項。

- 建議採用多可用區部署來防範可用區域故障。務必在多可用區架構中的叢集模式啟用的情況下進行部署，且最少有 3 個可用區域可使用。
- 建議使用全域資料存放區來防範區域故障。
- [最佳] 針對需要區域層級彈性的工作負載啟用全域資料存放區。
  - 制定計劃，以在主要區域降級時容錯移轉至次要區域。
  - 在生產環境中進行容錯移轉之前，先測試多區域容錯移轉程序。
  - 監控 ReplicationLag 指標，以了解容錯移轉事件期間資料遺失可能造成的影響。
- [資源]：
  - [減少故障](#)
  - [使用全域資料存放區跨 AWS 個區域進行複寫](#)
  - [從備份還原並選擇性地調整叢集規模](#)

• [利用多可用區將 ElastiCache for Redis 的停機時間降至最低](#)

## REL 4：如何有效地規劃容錯移轉？

**問題難易度簡介：**啟用多可用區的自動容錯移轉功能是 ElastiCache 的最佳實務。在某些情況下，ElastiCache for Redis 會在服務操作的過程中取代主節點。範例情況包括規劃的維護事件，以及少見的節點故障或可用區域問題。容錯移轉成功與否，取決於 ElastiCache 和您的用戶端程式庫組態。

**問題難易度優點：**遵循 ElastiCache 容錯移轉的最佳實務，並搭配您專屬的 ElastiCache for Redis 用戶端程式庫，有助於將容錯移轉事件期間可能的停機時間降到最低。

- [必要] 若叢集模式已停用，請使用逾時，如此用戶端就能偵測出是否需要與舊的主節點中斷連線，並使用更新的主要端點 IP 地址重新連線至新的主節點。若叢集模式已啟用，則用戶端程式庫會負責偵測基礎叢集拓撲中的變更。此作業最常透過 ElastiCache for Redis 用戶端程式庫中的組態設定來完成，您也可以在此組態設定中設定重新整理的頻率和方法。每個用戶端程式庫都提供自己的設定，如需詳細資訊，可參閱各自對應的文件。

[資源]：

- [利用多可用區將 ElastiCache for Redis 的停機時間降至最低](#)
- 檢閱 ElastiCache for Redis 用戶端程式庫的最佳實務。
- [必要] 容錯移轉成功與否，取決於主節點和複本節點之間是否有運作狀態良好的複寫環境。檢閱並了解 Redis 複寫的非同步性質，以及可用的 CloudWatch 指標，以通報主節點和複本節點之間的複寫延遲。針對需要更高資料安全性的使用案例，請利用 Redis WAIT 命令強制複本須先確認寫入，才能回應連線的用戶端。

[資源]：

- [Redis 的指標](#)
- [使用 Amazon CloudWatch 搭配 Amazon ElastiCache for Redis 進行監控的最佳實務](#)
- [最佳] 使用 ElastiCache 測試容錯移轉 API 在容錯移轉期間定期驗證應用程式的回應能力。

[資源]：

- [在 Amazon ElastiCache for Redis 上測試自動容錯移轉至讀取複本的程序](#)
- [測試自動容錯移轉](#)

## REL 5：您的 ElastiCache 元件是否設計為可擴展？

**問題難易度簡介：**藉由了解擴展功能和可用的部署拓撲，您的 ElastiCache 元件就可隨著時間進行調整，以因應不斷變化的工作負載需求。ElastiCache 提供 4 種擴展方式：縮減/橫向擴展 (水平) 以及上/下縱向擴展 (垂直)。

問題難易度優點：遵循 ElastiCache 部署的最佳實務可提供最大程度的擴展彈性，並符合水平擴展的 Well Architected 原則，進而將故障的影響降至最低。

- [必要] 了解叢集模式已啟用與叢集模式已停用的拓撲之間的差異。幾乎所有情況下都建議您在叢集模式已啟用時進行部署，因為這樣就能隨著時間提供更高的可擴展性。叢集模式已停用的元件會在藉由新增讀取複本進行水平擴展的能力上受到限制。
- [必要] 了解擴展的時機和方式。
  - 如需更多 READIOPS：新增複本
  - 如需更多 WRITEOPS：新增碎片 (橫向擴展)
  - 如需更多網路 IO：使用網路最佳化的執行個體，縱向擴展
- [最佳] 在叢集模式已啟用的情況下部署 ElastiCache 元件，並偏向部署更多、更小的節點，而不是較少、較大的節點。這樣做可有效地限制節點故障的影響範圍。
- [最佳] 在叢集中包含複本，以增強擴展事件期間的回應能力
- [良好] 若叢集模式已停用，請利用讀取複本來增加整體讀取容量。ElastiCache 在叢集模式已停用時最多可支援 5 個讀取複本，以及縱向擴展。
- [資源]：
  - [擴展 ElastiCache for Redis 叢集](#)
  - [線上縱向擴展](#)
  - [擴展 ElastiCache for Memcached 叢集](#)

## Amazon ElastiCache Well-Architected Lens 效能效率支柱

效能效率支柱著重於有效率地使用 IT 和運算資源。重要主題包括根據工作負載需求選取適當的資源類型和大小、監控效能，以及做出明智的決策，以便隨著業務需求的發展維持效率。

### 主題

- [PE 1：如何監控 Amazon ElastiCache 叢集的效能？](#)
- [PE 2：如何將工作分配到各個 ElastiCache 叢集節點？](#)
- [PE 3：針對快取工作負載，如何追蹤和報告快取的有效性和效能？](#)
- [PE 4：如何利用您的工作負載讓網路資源和連線得到最佳運用？](#)
- [PE 5：如何管理刪除和/或移出索引鍵？](#)
- [PE 6：如何在 ElastiCache 中建立資料模型並與資料互動？](#)
- [PE 7：如何記錄 Amazon ElastiCache 叢集中執行緩慢的命令？](#)

- [PE8：自動擴展功能如何協助提高 ElastiCache 叢集的效能？](#)

## PE 1：如何監控 Amazon ElastiCache 叢集的效能？

問題難易度簡介：藉由了解現有的監控指標，您就能識別目前的使用率。適當監控有助於識別影響叢集效能的潛在瓶頸。

問題難易度優點：了解與叢集相關聯的指標有助於引導最佳化技術，進而降低延遲並增加輸送量。

- [必要] 使用工作負載的子集進行基準效能測試。
  - 您應該使用像是負載測試這類機制來監控實際工作負載的效能。
  - 在執行這些測試的同時監控 CloudWatch 指標，便能對可用的指標有所了解，並且建立效能基準。
- [最佳] 針對 ElastiCache for Redis 工作負載，請重新命名運算上來說較為昂貴的命令，例如 KEYS，藉此限制使用者在生產叢集上執行封鎖命令的能力。
  - 執行引擎 6.x 的 ElastiCache for Redis 工作負載可利用角色型存取控制來限制特定命令。藉由在 AWS 主控台或 CLI 建立使用者和使用者群組，然後將使用者群組與 ElastiCache for Redis 叢集建立關聯，即可控制命令的存取權。在 Redis 6 中，當 RBAC 啟用時，我們可以使用 "-@dangerous"，這樣它就會禁止該使用者使用如 KEYS、MONITOR、SORT 等昂貴的命令。
  - 若是 5.x 版引擎，請使用 Amazon ElastiCache for Redis 叢集參數群組上的 `rename-commands` 參數重新命名命令。
- [較佳] 分析緩慢的查詢並尋找最佳化技術。
  - 針對 ElastiCache for Redis 工作負載，可透過分析慢速日誌來深入了解您的查詢。例如，您可以使用下列命令 `redis-cli slowlog get 10` 來顯示超出延遲閾值 (預設為 10 秒) 的最後 10 個命令。
  - 使用複雜的 ElastiCache for Redis 資料結構能夠更有效率地執行特定查詢。舉例來說，針對數值樣式範圍的查詢，應用程式可以使用排序集來實作簡單的數值索引。管理這些索引可以減少對資料集執行掃描的次數，並且以更高的效能效率傳回資料。
  - 針對 ElastiCache for Redis 工作負載，`redis-benchmark` 提供了簡單的介面，可利用像是用戶端數量和資料大小等使用者定義的輸入來測試不同命令的效能。
  - 由於 Memcached 僅支援簡單的索引鍵層級命令，因此請考慮建置其他索引鍵作為索引，以避免反覆查看索引鍵空間來處理用戶端查詢。
- [資源]：
  - [使用 CloudWatch 指標監控用量](#)
  - [使用 CloudWatch 指標監控用量](#)

- [使用 Amazon CloudWatch 警示](#)
- [Redis 專用參數](#)
- [SLOWLOG](#)
- [Redis 基準](#)

## PE 2：如何將工作分配到各個 ElastiCache 叢集節點？

問題難易度簡介：應用程式連線到 Amazon ElastiCache 節點的方式可能會影響叢集的效能和可擴展性。

問題難易度優點：正確使用叢集中的可用節點，可確保將工作分配到可用的資源。以下技術也有助於避免資源閒置。

- [必要] 讓用戶端連線到適當的 ElastiCache 端點。
  - Amazon ElastiCache for Redis 會根據使用中的叢集模式實作不同的端點。若叢集模式已啟用，ElastiCache 會提供組態端點。若叢集模式已停用，ElastiCache 會提供主要端點 (通常用於寫入) 和讀取器端點 (用於平衡複本之間的讀取)。正確實作這些端點可提升效能並且更輕鬆地擴展操作。除非有特定需求，否則避免連線到個別節點端點。
  - 針對多節點 Memcached 叢集，ElastiCache 提供了可進行自動探索的設定端點。建議使用雜湊演算法將工作平均分配到各個快取節點。許多 Memcached 用戶端程式庫會實作一致的雜湊。檢查您使用的程式庫文件，查看其是否支援一致性雜湊以及其實作方式。您可以在[這裡](#)找到有關實做這些功能的詳細資訊。
- [較佳] 實施策略來識別和修復工作負載中的快速鍵。
  - 請將多維度 Redis 資料結構的影響納入考量，例如清單、串流、集合等。這些資料結構存放在位於單一節點上的單一 Redis 索引鍵中。相當大型的多維度索引鍵可能比其他資料類型利用更多的網路容量和記憶體，並且可能造成該節點的使用率不成比例。您的工作負載設計應盡可能將資料存取分散到多個獨立的索引鍵。
  - 工作負載中的快速鍵可能會影響使用中節點的效能。對於 ElastiCache for Redis 工作負載，如果已設有 LFU 最大記憶體政策，則可以使用 `redis-cli --hotkeys` 偵測快速鍵。
  - 請考慮將快速鍵複寫到多個節點，讓分配到各節點的存取權更平均。此方法會要求用戶端寫入多個主要節點 (Redis 節點本身不會提供此功能)，而且除了原始索引鍵名稱之外，還要維護作為讀取來源的索引鍵名稱清單。
  - ElastiCache for Redis 第 6 版可支援伺服器輔助的[用戶端快取](#)。這可讓應用程式等待索引鍵變更，然後再回頭對 ElastiCache 進行網路呼叫。
- [資源]：

- [設定 Amazon ElastiCache for Redis 以提高可用性](#)
- [尋找連線端點](#)
- [負載平衡最佳實務](#)
- [Redis 中的用戶端快取](#)

### PE 3：針對快取工作負載，如何追蹤和報告快取的有效性和效能？

問題難易度簡介：快取是 ElastiCache 上常見的工作負載，因此您務必了解如何管理快取的有效性和效能。

問題難易度優點：您的應用程式可能會出現效能遲緩的跡象。假如能夠使用快取專用指標做出明智的決策來提高應用程式效能，這點對於快取工作負載而言至關重要。

- [必要] 測量並追蹤經過一段時間的快取命中率。快取的效率是由其「快取命中率」所決定。快取命中率的計算定義是索引鍵命中總數除以命中加未命中總數。命中率越接近 1，表示快取越有效。快取未命中數量是造成快取命中率低的原因。快取中找不到請求的索引鍵時，就會產生快取未命中數。快取中沒有某個索引鍵是因為該索引鍵已移出或刪除、已過期或不曾存在。了解索引鍵為什麼不在快取中，並制定適當的策略將其納入快取中。

[資源]：

- [必要] 測量並收集應用程式快取效能，並結合延遲和 CPU 使用率值，以了解是否需要調整存留時間或其他應用程式元件。ElastiCache 針對每一種資料結構提供了一組用於彙總延遲的 CloudWatch 指標。這些延遲指標是使用來自 ElastiCache for Redis INFO 命令的 commandstats 統計資料計算而得，且不包括網路和 I/O 時間。這只包括 ElastiCache for Redis 處理操作所花的時間。

[資源]：

- [使用 Amazon CloudWatch 搭配 Amazon ElastiCache for Redis 進行監控的最佳實務](#)
- [最佳] 選擇最合乎您需要的快取策略。快取未命中數量是造成快取命中率低的原因。如果您的工作負載設計是維持低快取未命中數量 (例如即時通訊)，那麼最好檢閱您的快取策略，並對工作負載套用最適當的解決方案 (例如查詢檢測) 來測量記憶體和效能。您實際用來填入和維護快取的策略，取決於您的用戶端需要快取的資料，以及該資料的存取模式。例如，您不太可能對串流應用程式上的個人化推薦和熱門新聞報導採用相同的策略。

[資源]：

- [快取策略](#)
- [快取最佳實務](#)

- [Performance at Scale with Amazon ElastiCache \(利用 Amazon ElastiCache 大規模提高效能\) 白皮書](#)

## PE 4：如何利用您的工作負載讓網路資源和連線得到最佳運用？

問題難易度簡介：許多應用程式用戶端都支援 ElastiCache for Redis 和 Memcached，但實作可能有所不同。您需要了解現有的網路和連線管理，以分析潛在的效能影響。

問題難易度優點：有效運用網路資源可改善叢集的效能效率。下列建議可減少網路需求，並改善叢集延遲和輸送量。

- [必要] 主動管理與您的 ElastiCache 叢集的連線。
  - 在應用程式中使用連線集區，可減少因開啟和關閉連線而對叢集造成的額外負荷量。在 Amazon CloudWatch 中使用 CurrConnections 和 NewConnections 監控連線行為。
  - 適時確實地關閉用戶端連線，以避免洩漏連線。連線管理策略包括確實地關閉未使用的連線，以及設定連線逾時。
  - 針對 Memcached 工作負載保留了用於處理連線的可設定記憶體數量，稱為 memcached\_connections\_overhead。
- [較佳] 壓縮大型物件以減少記憶體並改善網路輸送量。
  - 資料壓縮可減少所需的網路輸送量 (Gbps)，但會增加應用程式壓縮和解壓縮資料的工作量。
  - 壓縮還可以減少索引鍵耗用的記憶體數量
  - 請根據您應用程式的需求，考慮壓縮比和壓縮速度之間的權衡。
- [資源]：
  - [Amazon ElastiCache for Redis - 全域資料存放區](#)
  - [Memcached 專用參數](#)
  - [Amazon ElastiCache for Redis 5.0.3 增強了 I/O 處理能力以大幅提高效能](#)
  - [設定 Amazon ElastiCache for Redis 以提高可用性](#)

## PE 5：如何管理刪除和/或移出索引鍵？

問題難易度簡介：工作負載各有不同的需求，在叢集節點接近記憶體耗用限制時，也各有不同的預期行為。Amazon ElastiCache for Redis 會採用不同的政策來處理這些情況。

問題難易度優點：適當管理可用記憶體並且了解移出政策，將有助於確保在超過執行個體記憶體限制時，能夠察覺到叢集行為。

- [必要] 檢測資料存取權以評估要套用的政策。找出適當的最大記憶體政策，以控制是否要在叢集上執行移出，以及移出的方式。
  - 當達到叢集上的最大記憶體耗用量，而且已設有允許移出的政策時，就會進行移除。在此情況下，叢集的行為會取決於指定的移出政策。使用 ElastiCache for Redis 叢集參數群組上的 `maxmemory-policy` 即可管理此政策。
  - 預設政策 `volatile-lru` 會藉由將已設定到期時間 (TTL 值) 的索引鍵移出來釋出記憶體。最不常用 (LFU) 和最近最少使用 (LRU) 政策會根據使用情形移除索引鍵。
  - 針對 Memcached 工作負載，有一項預設的 LRU 政策可用來控制每個節點上的移出作業。您可以使用 Amazon CloudWatch 上的「移出」指標來監控 Amazon ElastiCache 叢集上的移出數量。
- [較佳] 將刪除行為標準化，就可控制對叢集的效能影響，進而避免非預期的效能瓶頸發生。
  - 針對 ElastiCache for Redis 工作負載，當明確從叢集中移除索引鍵時，`UNLINK` 就如 `DEL`：它會移除指定的索引鍵。然而，該命令會在不同的執行緒中執行實際的記憶體回收，因此不會封鎖，但 `DEL` 會封鎖。實際的移除操作會在之後以非同步方式進行。
  - 針對 ElastiCache for Redis 6.x 工作負載，`DEL` 命令的行為可在參數群組中使用 `lazyfree-lazy-user-del` 參數修改。
- [資源]：
  - [使用參數群組設定引擎參數](#)
  - [UNLINK](#)
  - [使用 AWS 進行雲端財務管理](#)

## PE 6：如何在 ElastiCache 中建立資料模型並與資料互動？

問題難易度簡介：ElastiCache 的應用相當依賴所使用的資料結構和資料模型，但同樣需要考慮基礎資料存放區 (如有的話)。了解可用的 ElastiCache for Redis 資料結構，並確保您使用最合乎您需要的資料結構。

問題難易度優點：ElastiCache 中的資料建模包含數層，包括應用程式使用案例、資料類型，以及資料元素之間的關係。此外，每一種 ElastiCache for Redis 資料類型和命令都各自擁有詳細記載的效能特色。

- [最佳] 最佳實務是減少意外覆寫資料的情況。使用盡可能減少索引鍵名稱重疊的命名慣例。資料結構的慣用命名是使用像是 `APPNAME:CONTEXT:ID` 這類階層方法，例如 `ORDER-APP:CUSTOMER:123`。

[資源]：

- [索引鍵命名](#)

- [最佳] ElastiCache for Redis 的命令具有採用 Big O 標記法定義的時間複雜度。命令的這種時間複雜度是其影響的演算/數學表示法。當您在應用程式中引入新的資料類型時，需仔細查看相關命令的時間複雜度。時間複雜度為  $O(1)$  的命令在時間上是恆定的，並不取決於輸入大小，但是時間複雜度為  $O(N)$  的命令在時間上是線性的，因此會受輸入大小的影響。由於 ElastiCache for Redis 採用單執行緒設計，因此大量高時間複雜度的操作將導致效能低落，並可能導致操作逾時。

[資源]：

- [命令](#)

- [最佳] 使用 API 來取得叢集中資料模型的 GUI 可見度。

[資源]：

- [Redis Commander](#)
- [Redis 瀏覽器](#)
- [Redsmin](#)

## PE 7：如何記錄 Amazon ElastiCache 叢集中執行緩慢的命令？

**問題難易度簡介：**透過擷取、彙總和通知長時間執行的命令，達到效能調整效益。藉由了解命令執行所需的時間，就可以判斷哪些命令會導致效能不佳，以及哪些命令造成引擎無法以最佳效能執行。Amazon ElastiCache for Redis 還能將這些資訊轉送至 Amazon CloudWatch 或 Amazon Kinesis Data Firehose。

**問題難易度優點：**針對慢速命令記錄到專用的永久位置並提供通知事件，有助於進行詳細的效能分析，並可用於觸發自動化事件。

- [必要] Amazon ElastiCache for Redis 執行 6.0 或更新版本的引擎、已正確設定參數群組，並已在叢集上啟用 SLOWLOG 記錄。
  - 必要的參數只有在引擎版本相容性設定為 Redis 6.0 或更高版本時才可用。
  - 當命令的伺服器執行時間超過指定的值時，SLOWLOG 記錄就會發生。叢集的行為取決於相關聯的參數群組參數，也就是 `slowlog-log-slower-than` 和 `slowlog-max-len`。
  - 變更會立即生效。
- [最佳] 利用 CloudWatch 或 Kinesis Data Firehose 的功能。
  - 使用 CloudWatch、CloudWatch Logs Insights 及 Amazon Simple Notification Service 的篩選和警示功能，就能實現效能監控和事件通知。

- 使用 Kinesis Data Firehose 的串流功能將 SLOWLOG 日誌封存到永久儲存體，或觸發自動化叢集參數調整。
- 判斷 JSON 或純文字格式最合乎您的需要。
- 提供 IAM 許可以發佈到 CloudWatch 或 Kinesis Data Firehose。
- [較佳] 將 `slowlog-log-slower-than` 設定為預設值以外的值。
  - 此參數決定命令可以在 Redis 引擎內執行的時間長度，經過這段時間後，該命令將會記錄為緩慢執行的命令。預設值為 10,000 微秒 (10 毫秒)。對於某些工作負載而言，預設值可能太高。
  - 根據應用程式需求和測試結果，決定更合乎您工作負載的值；不過，太低的值可能會產生過多資料。
- [較佳] 保留 `slowlog-max-len` 的預設值。
  - 此參數決定在任何特定時間於 Redis 記憶體中擷取的緩慢執行命令數目上限。值為 0 會有效停用擷取。值越高，記憶體中儲存的項目就越多，因而減少重要資訊在檢閱之前就遭到移出的情形。預設值為 128。
  - 預設值適用於大多數的工作負載。如果需要從 `redis-cli` 透過 SLOWLOG 命令分析一個較長時段的資料，請考慮增加此值。這樣就能在 Redis 記憶體中保留更多命令。

如果您要將 SLOWLOG 資料發送到 CloudWatch Logs 或 Kinesis Data Firehose，資料將會保留下來，並可在 ElastiCache 系統之外進行分析，如此就能減少在 Redis 記憶體中儲存大量緩慢執行命令的需求。

- [資源]：
  - [如何在 ElastiCache for Redis 快取叢集中開啟 Redis 慢速日誌？](#)
  - [日誌傳送](#)
  - [Redis 專用參數](#)
  - <https://aws.amazon.com/cloudwatch/> Amazon CloudWatch
  - [Amazon Kinesis Data Firehose](#)

## PE8：自動擴展功能如何協助提高 ElastiCache 叢集的效能？

問題難易度簡介：透過實作 Redis 自動擴展功能，您的 ElastiCache 元件就能隨著時間進行調整，以自動增加或減少所需的碎片或複本。藉由實作目標追蹤或排程的擴展政策就能達到此目的。

問題難易度優點：了解並規劃工作負載的尖峰，就可確實增強快取效能與業務連續性。ElastiCache for Redis 自動擴展功能會持續監控您的 CPU/記憶體使用率，確保叢集依照您所需的效能層級運作。

- [必要] 啟動 ElastiCache for Redis 的叢集時：

1. 確定已啟用叢集模式
2. 確定執行個體屬於支援自動擴展的特定類型和大小系列
3. 確保叢集未在全域資料存放區、Outposts 或 Local Zones 中執行

[資源]：

- [擴展 Redis \(叢集模式已啟用\) 中的叢集](#)
  - [搭配碎片使用自動擴展](#)
  - [搭配複本使用自動擴展](#)
- [最佳] 確認您的工作負載為大量讀取或大量寫入，以定義擴展政策。為了獲得最佳效能，請使用單獨一個追蹤指標。建議您避免針對每個維度實施多項政策，因為自動擴展政策會在命中目標時橫向擴展，但只有在所有目標追蹤政策都準備好縮減時才會縮減。

[資源]：

- [自動擴展政策](#)
  - [定義擴展政策](#)
- [最佳] 隨著時間監控效能可幫助您偵測到在特定時間點監控時，可能無法察覺的工作負載變更。您可以分析對應的 CloudWatch 指標來了解四週期間內的叢集使用率，藉此判斷目標值的閾值。如果您仍不確定要選擇哪個值，建議從支援的最小預先定義指標值開始。

[資源]：

- [使用 CloudWatch 指標監控用量](#)
- [較佳] 我們建議您使用預期的最小和最大工作負載來測試應用程式，藉此找出叢集制定擴展政策和減輕可用性問題所需的準確碎片/複本數量。

[資源]：

- [註冊可擴展的目標](#)
- [註冊可擴展的目標](#)

## Amazon ElastiCache Well-Architected Lens 成本最佳化支柱

成本最佳化支柱著重於避免不必要的成本。重要主題包括了解和控制錢花費在哪裡、選取最合適的節點類型 (使用支援根據工作負載需求進行資料分層的執行個體)、適切的資源類型數量 (有多少個讀取複本)、分析經過一段時間後的花費，以及因應業務需求進行擴展而不超支。

- [COST 1：如何找出並追蹤與 ElastiCache 資源相關聯的成本？如何制定出各種機制，讓使用者能夠建立、管理及處置建立的資源？](#)
- [COST 2：如何使用持續監控工具來協助您將有關 ElastiCache 資源的成本最佳化？](#)
- [COST 3：是否應該使用支援資料分層的執行個體類型？資料分層有何優點？何時不適合使用資料分層執行個體？](#)

**COST 1：如何找出並追蹤與 ElastiCache 資源相關聯的成本？如何制定出各種機制，讓使用者能夠建立、管理及處置建立的資源？**

問題難易度簡介：若要了解成本指標，就需要有多個團隊參與並且跨團隊協作：軟體工程、資料管理、產品負責人、財務及領導階層。若要找出關鍵的成本驅動因素，就需要參與的各方都了解服務用量控制槓桿與成本管理權衡，這時常成為投入成本最佳化的努力獲得成功與不太成功的關鍵差異。確保您擁有現成的程序和工具可追蹤從開發到生產與淘汰整個過程中所建立的資源，便能幫助您管理 ElastiCache 伴隨的成本。

問題難易度優點：若要持續追蹤與工作負載相關聯的所有成本，便需要深入了解包含 ElastiCache 這個元件在內的架構。此外，您應制定成本管理計劃來收集用量，並與您的預算進行比較。

- [必要] 建構雲端卓越中心 (CCoE)，並將自行定義、追蹤有關組織的 ElastiCache 使用量指標，以及對指標採取行動等納入，成為其中一篇創始章程。如果 CCoE 已存在且運作正常，請確定它知道如何讀取和追蹤與 ElastiCache 相關的成本。建立資源時，請使用 IAM 角色和政策來確認只有特定團隊和群組能夠執行個體化資源。這樣可確保成本與業務成果相關聯，並從成本的角度建立明確的責任脈絡。
  1. CCoE 應識別、定義並發佈成本指標，這些指標會根據分類資料中重要的 ElastiCache 使用量定期 (每月) 更新，例如：
    - a. 使用的節點類型及其屬性：標準與記憶體最佳化、隨需與預留執行個體、區域和可用區域
    - b. 環境類型：免費、開發、測試和生產
    - c. 備份儲存與保留策略
    - d. 區域內與跨區域的資料傳輸
    - e. 在 Amazon Outposts 上執行的執行個體
  2. CCoE 是由一個跨職能的團隊所組成，其中包括您組織中軟體工程、資料管理、產品團隊、財務和領導團隊的非專屬代表。

[資源]：

- [打造雲端卓越中心](#)

- [Amazon ElastiCache 定價](#)

- [必要] 使用成本分配標籤以較低精細程度來追蹤成本。使用 AWS Cost Management 來視覺化、了解並管理隨時間推移的 AWS 成本和使用量。
  1. 使用標籤來整理您的資源，而成本分配標籤可以用來深入追蹤 AWS 成本的細節。啟用成本分配標籤後，AWS 會使用成本分配標籤在成本分配報告上整理您的資源成本，讓您輕鬆分類和追蹤 AWS 成本。AWS 提供兩種類型的成本分配標籤：AWS 產生的標籤和使用者定義的標籤。AWS 會為您定義、建立和套用 AWS 產生的標籤，而您負責定義、建立和套用使用者定義的標籤。您必須分別啟用這兩種標籤，它們才會顯示在 Cost Management 或成本分配報告中。
  2. 您可以使用成本分配標籤來整理您的 AWS 帳單，以反映您自己的成本結構。若將成本分配標籤新增至 Amazon ElastiCache 中的資源，您就可依資源標籤值將發票上的費用分組，藉此追蹤成本。您可考慮結合標籤，以便更深入追蹤成本的細節。

[資源]：

- [使用 AWS 成本分配標籤](#)
- [使用成本分配標籤監控成本](#)
- [AWS Cost Explorer](#)
- [最佳] 將 ElastiCache 成本連結到整個組織的指標。
  1. 考慮業務指標以及像是延遲等操作指標 - 您的業務模型中有哪些概念是可以跨角色理解的？這些指標需要讓組織中越多角色理解越好。
  2. 範例 - 同時提供服務的使用者、每項操作和每個使用者的最大和平均延遲、使用者參與度分數、使用者回流率/週、工作階段長度/使用者、放棄率、快取命中率，以及追蹤的索引鍵

[資源]：

- [使用 CloudWatch 指標監控用量](#)
- [良好] 在使用 ElastiCache 的整個工作負載中，維持指標與成本的最新架構和操作能見度。
  1. 了解您的整個解決方案生態系統就能得知，ElastiCache 通常是其技術集之中，從用戶端到 API 閘道、Redshift 和 QuickSight 完整的 AWS 服務生態系統的一部分，舉例來說，它可做為報告工具。
  2. 在您的架構圖上，對應來自用戶端、連線、安全性、記憶體內操作、儲存、資源自動化、資料存取和管理的解決方案元件。每一層都連接到整個解決方案，並且有自己的需求和功能，能夠增添和/或有助您管理整體成本的能力。
  3. 您的圖表應包括使用運算、網路、儲存、生命週期政策、指標收集，以及應用程式的操作和運作 ElastiCache 元素

4. 工作負載的需求可能會隨著時間而演進，因此您務必繼續維護並記錄對基礎元件以及主要功能目標的理解程度，以便在工作負載成本管理中保持主動。
5. 為了針對 ElastiCache 制定出有效的成本管理策略，高階主管對於可見性、責任、優先順序和資源的支援對您來說便至關重要。

## COST 2：如何使用持續監控工具來協助您將有關 ElastiCache 資源的成本最佳化？

**問題難易度簡介：**您需要在 ElastiCache 成本和應用程式效能指標之間取得適當的平衡。Amazon CloudWatch 提供關鍵操作指標的可見性，能夠協助您根據需求評估 ElastiCache 資源利用是否過度或不足。從成本最佳化的觀點來看，您需要了解何時過度佈建，並能夠制定出適當的機制來調整 ElastiCache 資源的規模，同時維持您的運作、可用性、彈性和效能需求。

**問題難易度優點：**在理想狀態下，您將佈建足夠的資源來滿足工作負載運作需求，並且沒有資源使用率不足的情況，而導致處於非最佳成本狀態。您需要能夠識別並避免長時間操作規模過大的 ElastiCache 資源。

- [必要] 使用 CloudWatch 監控您的 ElastiCache 叢集，並分析這些指標與您的 AWS Cost Explorer 儀表板之間的關聯性。
  1. ElastiCache 提供主機層級指標 (例如 CPU 用量) 及快取引擎軟體的特定指標 (例如快取取得及快取遺失)。每隔 60 秒會針對每個快取節點測量及發佈這些指標。
  2. ElastiCache 效能指標 (CPUUtilization、EngineUtilization、SwapUsage、CurrConnections 及 Evictions) 可指出您需要縱向擴展/縮減 (使用較大/較小的快取節點類型) 或橫向擴展/縮減 (增加/減少碎片)。藉由建立教戰手冊對照表來預估額外成本，以及達到應用程式效能閾值所需的最短和最長時間，從而了解擴展決策的成本影響。

[資源]：

- [使用 CloudWatch 指標監控用量](#)
- [應監控哪些指標？](#)
- [Amazon ElastiCache 定價](#)
- [必要] 了解並記錄您的備份策略和成本影響。
  1. 使用 ElastiCache 時，備份會存放在 Amazon S3 中以便長久儲存。您需要了解與您的故障復原能力相關的成本影響。
  2. 啟用自動備份，這樣將會刪除保留期限已過的備份檔案。

[資源]：

- [排程自動備份](#)

- [Amazon Simple Storage Service 定價](#)
- [最佳] 針對執行個體使用預留節點，這是為了管理已充分了解並記錄的工作負載成本刻意而為的策略。您必須先為預留節點預付費用，實際費用取決於節點類型及保留時間長度（一或三年）。此費用遠低於您使用隨需節點時需支付的每小時使用費。
  1. 您可能需要使用隨需節點操作 ElastiCache 叢集，直到您收集了足夠的資料來預估預留執行個體的需求。規劃並記錄滿足您的需求所需的資源，並比較各執行個體類型（隨需與預留）的預期成本
  2. 定期評估可用的新快取節點類型，並從成本和操作指標的角度評估是否合理，以便將您的執行個體機群移轉到新的快取節點類型

### COST 3：是否應該使用支援資料分層的執行個體類型？資料分層有何優點？何時不適合使用資料分層執行個體？

問題難易度簡介：選取適當的執行個體類型不僅會影響效能和服務層面，還會影響財務層面。執行個體類型有各種不同的相關成本。您可能自然而然會選取一個或少數幾個可滿足記憶體中所有儲存需求的大型執行個體類型。但是隨著專案逐漸成熟，這可能會產生重大的成本影響。為了確保選取正確的執行個體類型，就需要定期檢查 ElastiCache 物件閒置時間。

問題難易度優點：您應該清楚了解各種不同的執行個體類型對您目前和未來的成本有何影響。邊際或定期工作負載變更不應造成不成比例的成本變更。在工作負載允許的情況下，選擇支援資料分層的執行個體類型就能提供單一儲存價格更實惠的可用儲存。這是因為每個執行個體可用的 SSD 儲存資料分層執行個體可支援的單一執行個體總資料量高出許多。

- [必要] 了解資料分層執行個體的限制
  1. 僅適用於 ElastiCache for Redis 叢集。
  2. 僅限幾種執行個體類型可支援資料分層。
  3. 僅支援 ElastiCache for Redis 6.2 及更高版本
  4. 大型項目不會交換至 SSD。超過 128 MiB 的物件會保存在記憶體中。

[資源]：

- [資料分層](#)
- [Amazon ElastiCache 定價](#)
- [必要] 了解工作負載定期存取的資料庫百分比。
  1. 資料分層執行個體非常適合經常存取整個資料集的一小部分，但仍需要快速存取其餘資料的工作負載。換句話說，熱資料對暖資料的比例約為 20:80。
  2. 發展叢集層級的物件閒置時間追蹤。

3. 超過 500 Gb 資料量的大型實作就是很好的選擇。
- [必要] 了解資料分層執行個體並非某些工作負載的可選項目。
    1. 存取較少用的物件會伴隨很小的效能成本，因為這些物件會交換至本機 SSD。如果您的應用程式對回應時間較為敏感，請測試對工作負載的影響。
    2. 不適用大多儲存大小超過 128 MiB 之大型物件的快取。

[資源]：

- [限制](#)
- [最佳] 預留執行個體類型可支援資料分層。這樣可確保每個執行個體的資料儲存量擁有最低成本。
  1. 在您充分了解自己的需求之前，可能需要使用非資料分層執行個體操作 ElastiCache 叢集。
  2. 分析您的 ElastiCache 叢集資料使用模式。
  3. 建立自動化的工作來定期收集物件閒置時間。
  4. 如果您發現有很大百分比 (約 80%) 的物件閒置了一段時間，則視為適合讓您的工作負載記錄調查結果，並建議將叢集遷移到支援資料分層的執行個體。
  5. 定期評估可用的新快取節點類型，並從成本和操作指標的角度評估是否合理，以便將您的執行個體機群移轉到新的快取節點類型。

[資源]：

- [OBJECT IDLETIME](#)
- [Amazon ElastiCache 定價](#)

## 常見疑難排解步驟和最佳做法

主題

- [連線問題](#)
- [Redis 的客戶端錯誤](#)
- [疑難排解 ElastiCache 無伺服器中的高延遲](#)
- [疑難排解無伺服器中的節流問題 ElastiCache](#)
- [相關主題](#)

## 連線問題

如果您無法連線到 ElastiCache 快取記憶體，請考慮下列其中一項：

1. 使用 TLS：如果您在嘗試連接到 ElastiCache 端點時遇到掛斷連接，則可能沒有在客戶端中使用 TLS。如果您使用的是 ElastiCache 無伺服器，傳輸中的加密一律會啟用。確定您的用戶端使用 TLS 連線至快取。在這裡進一步了解如何連線至啟用 TLS [的快取](#)。
2. VPC：ElastiCache 快取只能從 VPC 內存取。確保您從中存取快取和快取的 EC2 執行個體是在 ElastiCache 相同的 VPC 中建立的。或者，您必須在 EC2 執行個體所在的 VPC 和建立快取的 VPC 之間啟用 VPC [對等](#)。
3. 安全性群組：ElastiCache 使用安全性群組來控制快取的存取。考慮下列各項：
  - a. 確保您的 ElastiCache 緩存使用的安全組允許從 EC2 實例對其進行入站訪問。請參閱[此處](#)以瞭解如何正確設定安全性群組中的輸入規則。
  - b. 請確定快取所使用的安全性群組允許存取 ElastiCache 快取的連接埠 (無伺服器則為 6379 和 6380，自行設計的預設為 6379 和 6380)。ElastiCache 使用這些端口來接受 Redis 命令。[在這裡](#)進一步了解如何設定連接埠存取。

## Redis 的客戶端錯誤

ElastiCache 無伺服器只能使用支援 Redis 叢集模式通訊協定的 Redis 用戶端存取。根據叢集配置，可以從 Redis 用戶端以任一模式存取自行設計的叢集。

如果您在用戶端遇到 Redis 錯誤，請考慮下列事項：

1. 叢集模式：如果您遇到 CROSSLOT 錯誤或 SELE [CT](#) Redis 命令發生錯誤，您可能會嘗試使用不支援 Redis 叢集通訊協定的 Redis 用戶端來存取已啟用叢集模式的快取。ElastiCache 無伺服器僅支援支援 Redis 叢集通訊協定的 Redis 用戶端。如果要在「群集模式禁用」(CMD) 中使用 Redis，則必須設計自己的集群。
2. CROSSLOT 錯誤：如果您遇到 ERR CROSSLOT Keys in request don't hash to the same slot 錯誤，您可能正在嘗試存取不屬於叢集模式快取中相同插槽的金鑰。提醒您，ElastiCache 無伺服器一律以叢集模式運作。只有當涉及的所有密鑰都在相同的哈希插槽中時，才允許涉及多個密鑰的多鍵操作，事務或 Lua 腳本。

如需設定 Redis 用戶端的其他最佳作法，請參閱此[部落格文章](#)。

## 疑難排解 ElastiCache 無伺服器中的高延遲

如果您的工作負載出現高延遲，您可以分析 CloudWatch `SuccessfulReadRequestLatency` 和 `SuccessfulWriteRequestLatency` 指標，以檢查延遲是

否與 ElastiCache 無伺服器相關。這些指標會測量 ElastiCache 無伺服器內部的延遲時間-不包括用戶端和 ElastiCache 無伺服器端點之間的用戶端延遲和網路行程時間。

## 疑難排解用戶端

如果您注意到用戶端的延遲提高，但沒有相應的增加 CloudWatch `SuccessfulReadRequestLatency` 和測量伺服器端延遲的 `SuccessfulWriteRequestLatency` 指標，請考慮下列事項：

- 確定安全性群組允許存取連接埠 6379 和 6380：ElastiCache 無伺服器使用 6379 連接埠作為主要端點，而 6380 連接埠則用於讀取器端點。某些用戶端會為每個新連線建立兩個連接埠的連線，即使您的應用程式未使用「從複本讀取」功能也一樣。如果您的安全性群組不允許對這兩個連接埠進行輸入存取，則建立連線可能需要更長的時間。[在這裡](#)進一步了解如何設定連接埠存取。

## 疑難排解伺服器

一些變異性和偶爾的尖峰不應該是引起關注的原因。但是，如果 Average 統計數據顯示急劇增加並持續存在，則應檢查 AWS Health Dashboard 和您的 Personal Health Dashboard 以獲取更多信息。如有必要，請考慮使用 AWS Support。

請考慮下列減少延遲的最佳做法和策略：

- 啟用從複本讀取：如果您的應用程式允許，建議您在 Redis 用戶端中啟用「從複本讀取」功能來擴展讀取並達到更低的延遲。啟用時，ElastiCache 無伺服器會嘗試將讀取請求路由到與用戶端位於相同可用區域 (AZ) 的複本快取節點，以避免跨可用區域網路延遲。請注意，在用戶端中啟用「從複本讀取」功能表示您的應用程式會接受資料的最終一致性。如果您在寫入金鑰後嘗試讀取，您的應用程式可能會在一段時間內收到較舊的資料。
- 確保您的應用程序部署在與緩存相同的 AZ 中：如果您的應用程序未部署在與緩存相同的 AZ 中，則可能會發現更高的客戶端延遲。建立無伺服器快取時，您可以提供應用程式存取快取的子網路，而 ElastiCache 無伺服器會在這些子網路中建立 VPC Endpoint。確定您的應用程式部署在相同的 AZ 中。否則，您的應用程式可能會在存取快取時產生跨 AZ 躍點，從而導致用戶端延遲較高。
- 重複使用連線：ElastiCache 無伺服器要求是透過啟用 TLS 的 TCP 連線，使用 RESP 通訊協定進行。啟動連接（包括驗證連接，如果配置）需要一些時間，因此第一個請求的延遲比典型更高。透過已初始化連線的要求可提供一致 ElastiCache 的低延遲。因此，您應該考慮使用連線集區或重複使用現有的 Redis 連線。
- 擴展速度：ElastiCache 無伺服器會隨著您的請求率增加而自動擴展。如果要求率突然大幅增加，比 ElastiCache 無伺服器擴充速度快，可能會導致延遲提高一段時間。ElastiCache 無伺服器通常可以快速提高支援的要求率，最多需要 10 到 12 分鐘才能將請求率提高一倍。

- 檢查長時間運行的命令：某些 Redis 命令，包括 Lua 腳本或大型數據結構上的命令，可能會運行很長時間。若要識別這些指令，請 ElastiCache 發佈指令層級度量。使用 [ElastiCache 無伺服器](#)，您可以使用指BasedECPUs標。
- 限制要求：在 ElastiCache 無伺服器中限制要求時，您可能會遇到應用程式用戶端延遲增加的情況。[在 ElastiCache 無伺服器中限制要求時，您應該會看到無伺服器指標有所增加ThrottledRequestsElastiCache](#)。請檢閱以下章節以進行節流請求的疑難排解。
- 密鑰和請求的統一分佈：在 ElastiCache Redis 中，每個插槽的密鑰或請求分配不均勻可能導致熱插槽，從而導致延遲提高。ElastiCache 在執行簡單 SET/GET 命令的工作負載中，無伺服器在單一插槽上最多支援每秒 30,000 個 ECP (使用「從複本讀取」時/秒為 90,000 個 ECP)。我們建議您評估金鑰和跨插槽的要求分配，並在您的請求率超過此限制時確保一致的分配。

## 疑難排解無伺服器中的節流問題 ElastiCache

在服務導向架構和分散式系統中，限制各種服務元件處理 API 呼叫的速率稱為限流。這樣可以平滑尖峰、控制元件輸送量的不相符，並在發生意外的操作事件時提供更可預測的復原。ElastiCache 無伺服器專為這些類型的架構而設計，而且大多數 Redis 用戶端針對節流請求都內建了重試。某種程度的限流對應用程式來說不一定是問題，但資料工作流程中對延遲敏感的部分如果持續遭到限流，可能會對使用者體驗造成負面影響，並降低系統的整體效率。

[在 ElastiCache 無伺服器中限制要求時，您應該會看到無伺服器指標有所增加ThrottledRequestsElastiCache](#)。如果您注意到大量限制的要求，請考慮下列事項：

- 擴展速度：ElastiCache 無伺服器會隨著您擷取更多資料或提高請求速率而自動擴展。如果您的應用程式擴展速度快於無伺服器擴展的速度，則 ElastiCache 無伺服器擴展以適應您的工作負載時 ElastiCache，您的請求可能會受到限制。ElastiCache 無伺服器通常可以快速增加儲存大小，最多需要 10-12 分鐘才能將快取中的儲存大小增加一倍。
- 密鑰和請求的統一分佈：ElastiCache 對於 Redis，每個插槽的密鑰或請求分配不均勻可能導致熱插槽。如果在執行簡單 SET/GET 命令的工作負載中，單一插槽的要求速率超過每秒 30,000 ECP，熱插槽可能會導致請求節流。
- 從複本讀取：如果應用程式允許，請考慮使用「從複本讀取」功能。大多數 Redis 用戶端都可以設定為「縮放讀取」，以將讀取導向複本節點。此功能可讓您擴展讀取流量。此外，ElastiCache 無伺服器會自動將讀取從複本請求路由到與應用程式相同的可用區域中的節點，從而降低延遲。啟用「從複本讀取」時，對於使用簡單 SET/GET 命令的工作負載，您可以在單一插槽上達到最多 90,000 個 ECP。

## 相關主題

- [其他疑難排解步](#)
- [the section called “最佳實務和快取策略”](#)

## 其他疑難排解步

疑難排解持續連線問題時，必須驗證下列項目 ElastiCache：

### 主題

- [安全群組](#)
- [網路 ACL](#)
- [路由表](#)
- [DNS 解析](#)
- [透過伺服器端診斷識別問題](#)
- [網路連線能力驗證](#)
- [網路相關限制](#)
- [CPU 使用率](#)
- [從伺服器端終止連線](#)
- [Amazon EC2 執行個體的用戶端疑難排解](#)
- [解析完成單個請求所花費的時間](#)

## 安全群組

安全群組是保護 ElastiCache 用戶端 (EC2 執行個體、AWS Lambda 函數、Amazon ECS 容器等) 和 ElastiCache 快取的虛擬防火牆。安全群組具有狀態，這表示允許傳入或傳出流量之後，該流量的回應將在該特定安全群組的內容中自動授權。

具狀態功能需要安全群組追蹤所有授權的連線，而且追蹤的連線數有限額。如果達到限額，新的連線將會失敗。請參閱故障排除部分，以獲取有關如何識別用戶端或 ElastiCache 端限制是否已達到的幫助。

您可以同時將單一安全群組指派給用戶端和 ElastiCache 叢集，也可以指派給每個群組的個別安全群組。

對於這兩種情況，您都必須允許來自來源的 ElastiCache 連接埠上的 TCP 輸出流量，以及相同連接埠上的輸入流量允許到 ElastiCache。Memcached 的預設連接埠是 11211，Redis 則為 6379。根據預設，安全群組允許所有對外流量。在此情況下，只需要目標安全群組中的傳入規則。

如需詳細資訊，請參閱 [在 Amazon VPC 中存取 ElastiCache 叢集的存取模式](#)。

## 網路 ACL

網路存取控制清單 (ACL) 是無狀態規則。必須允許雙向 (傳入和傳出) 的流量才能成功。網路 ACL 會指派給子網路，而非特定資源。您可以將相同的 ACL 指派給 ElastiCache 和用戶端資源，特別是當它們位於相同的子網路時。

根據預設，網路 ACL 允許所有流量。不過，您可以自訂為拒絕或允許流量。此外，ACL 規則的評估會依序進行，這表示符合流量的數字最小規則將會予以允許或拒絕。允許 Redis 流量的最小組態是：

用戶端網路 ACL：

- 傳入規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- Type (類型)：Custom TCP Rule (自訂 TCP 規則)；
- Protocol (通訊協定)：TCP
- Port Range (連接埠範圍)：1024-65535
- 來源：0.0.0.0/0 (或為叢集子網路建立個別規則) ElastiCache
- Allow/Deny (允許/拒絕)：Allow (允許)
  
- 傳出規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- Type (類型)：Custom TCP Rule (自訂 TCP 規則)；
- Protocol (通訊協定)：TCP
- Port Range (連接埠範圍)：6379
- 資料來源：0.0.0.0/0 (或叢集子網路。ElastiCache 請記住，在容錯移轉或擴展叢集的情況下，使用特定 IP 可能會產生問題)
- Allow/Deny (允許/拒絕)：Allow (允許)

ElastiCache 網路 ACL：

- 傳入規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- Type (類型)：Custom TCP Rule (自訂 TCP 規則)；
- Protocol (通訊協定)：TCP
- Port Range (連接埠範圍)：6379
- 來源：0.0.0.0/0 (或為叢集子網路建立個別規則) ElastiCache
- Allow/Deny (允許/拒絕)：Allow (允許)
  
- 傳出規則：
- Rule number (規則編號)：最好低於任何拒絕規則；
- Type (類型)：Custom TCP Rule (自訂 TCP 規則)；
- Protocol (通訊協定)：TCP
- Port Range (連接埠範圍)：1024-65535
- 資料來源：0.0.0.0/0 (或叢集子網路。ElastiCache 請記住，在容錯移轉或擴展叢集的情況下，使用特定 IP 可能會產生問題)
- Allow/Deny (允許/拒絕)：Allow (允許)

如需詳細資訊，請參閱[網路 ACL](#)。

## 路由表

類似於網路 ACL，每個子網路也可以有不同的路由表。如果用戶端和 ElastiCache 叢集位於不同的子網路中，請確定其路由表允許它們互相連接。

較複雜的環境 (涉及多個 VPC、動態路由或網路防火牆) 可能會變得難以疑難排解。請參閱[網路連線能力驗證](#)，確認您的網路設定是否適當。

## DNS 解析

ElastiCache 根據 DNS 名稱提供服務端點。可用的端點為 Configuration、Primary、Reader 及 Node 端點。如需詳細資訊，請參閱[尋找連線端點](#)。

在修改容錯移轉或叢集的情況下，與端點名稱相關聯的地址可能會變更且會自動更新。

自訂 DNS 設定 (即不使用 VPC DNS 服務) 可能不知道 ElastiCache 提供的 DNS 名稱。確保您的系統可以使用系統工具 dig (如下所示) 或成功解析 ElastiCache 端點 nslookup。

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com
example-001.xxxxxx.0001.use1.cache.amazonaws.com.
1.2.3.4
```

您也可以透過 VPC DNS 服務強制執行名稱解析：

```
$ dig +short example.xxxxxx.ng.0001.use1.cache.amazonaws.com @169.254.169.253
example-001.tihewd.0001.use1.cache.amazonaws.com.
1.2.3.4
```

## 透過伺服器端診斷識別問題

CloudWatch 來自 ElastiCache 引擎的指標和執行階段資訊是識別潛在連線問題來源的常見來源或資訊。優質的分析資料通常從下列項目開始：

- CPU 使用率：Redis 是多執行序應用程式。但是每個命令的執行發生在單個 (主) 執行序中。基於這個原因，請 ElastiCache 提供指標 CPUUtilization 和 EngineCPUUtilization。EngineCPUUtilization 提供 Redis 處理序專用的 CPU 使用率，以及所有 vCPUs CPUUtilization 的使用率。具有多個 vCPU 的節點的 CPUUtilization 和 EngineCPUUtilization 值一般會不同，後者通常較高。出現高 EngineCPUUtilization，可能是由於需要大量 CPU 時間才能完成的請求或複雜作業數量增加所造成。您可以透過以下項目來識別兩者：
  - 請求數增加：檢查其他符合 EngineCPUUtilization 模式的指標是否增加。有用的指標包括：
    - CacheHits 和 CacheMisses：成功請求或沒有在快取中找到有效項目的請求之數目。如果未命中與命中的比率很高，表示應用程式正在對無效果的請求浪費時間和資源。
    - SetTypeCmds 和 GetTypeCmds：這些指標與 EngineCPUUtilization 相關，有助於了解寫入請求 (由 SetTypeCmds 測量) 或讀取請求 (由 GetTypeCmds 測量) 的負載是否明顯高。如果負載主要是讀取作業，使用多個僅供讀取複本可以在多個節點之間平衡請求，並撥出主節點用於寫入作業。在停用叢集模式的叢集中，您可以使用讀取器端點在應用程式中建立額外的連線組態來完成僅供讀取複本的使用。ElastiCache 如需詳細資訊，請參閱 [尋找連線端點](#)。讀取作業必須提交至這個額外的連線。寫入作業將透過一般主要端點完成。在啟用叢集模式的情況下，建議您使用原生支援僅供讀取複本的程式庫。若使用正確的標記，程式庫將能自動探索叢集拓撲、複本節點、透過 [READONLY](#) Redis 命令啟用讀取作業，並將讀取請求提交給複本。
- 連線數增加：
  - CurrConnections 和 NewConnections：CurrConnection 是收集資料點時的已建立連線數，而 NewConnections 會顯示這段期間內建立的連線數。

建立和處理連線意味著可觀的 CPU 額外負荷。此外，成立新連線所需的 TCP 三向交握會對整體回應時間產生負面影響。

NewConnections 每分鐘數千個 ElastiCache 節點表示連接僅由幾個命令創建和使用，這是不是最佳的。保持連線的成立狀態，並重複使用這些連線來進行新作業是最佳實務。當用戶端應用程式支援並妥善實作連線集區或持續連線時，此情況有可能實現。使用連線集區時，currConnections 數不會有很大的變化，而 NewConnections 應盡可能壓低。Redis 透過少量的 currConnection 提供最佳效能。讓 CurrConnection 數量維持在大約數十或數百個，可最大限度壓低資源的使用量，以支援個別連線，例如供連線使用的用戶端緩衝區和 CPU 週期。

- 網路輸送量：

- 判斷頻寬：ElastiCache 節點的網路頻寬與節點大小成比例。由於應用程式具有不同的特性，結果可能會根據工作負載而有所不同。舉例來說，小型請求率高的應用程式，對 CPU 使用率造成的影響往往高於網路輸送量，而索引鍵較大則會造成網路使用率較高。因此，建議您使用實際工作負載來測試節點，以便更深入了解限制。

模擬來自應用程式的負載會提供更準確的結果。但是，基準化分析工具可讓您更妥善了解限制。

- 針對請求主要都是讀取的情況，使用複本進行讀取作業將能減輕主節點上的負載。如果使用案例主要是寫入，則使用許多複本將會提高網路使用量。針對寫入主節點的每個位元組，系統會將 N 個位元組傳送到複本 (N 即複本的數量)。寫入密集型工作負載的最佳做法是 ElastiCache 針對具有叢集模式功能的 Redis 使用，因此可以在多個碎片之間平衡寫入，或擴展為具有更多網路功能的節點類型。
- CloudWatchmetrics NetworkBytesIn 並分別 NetworkBytesOut 提供進入或離開節點的資料量。ReplicationBytes 是專用於資料複製的流量。

如需詳細資訊，請參閱 [網路相關限制](#)。

- 複雜命令：Redis 命令在單個執行序上提供，這表示會按順序提供請求。單一緩慢命令可能會影響其他請求和連線，最終導致逾時。使用作用於多個值、索引鍵或資料類型的命令時必須小心。根據參數的數量或其輸入或輸出值的大小，可能會封鎖或終止連線。

一個惡名昭彰的例子是 KEYS 命令。它會掃描整個 Keyspace，搜尋指定的模式，並在其執行過程中阻止其他命令的執行。Redis 使用「大 O」標記法來描述其命令的複雜性。

索引鍵命令帶有  $O(N)$  時間複雜度，N 是資料庫中索引鍵的數量。因此，索引鍵數越多，命令執行速度就會越慢。KEYS 可能會以不同的方式造成問題：如果沒有使用搜尋模式，該命令將會傳回所

有可用的索引鍵名稱。在具有數千個或數百萬個項目的資料庫中，系統會建立巨大的輸出並淹沒網路緩衝區。

如果使用搜尋模式，則只會將符合該模式的索引鍵傳回用戶端。但是，引擎仍然會掃描整個 Keyspace 搜尋它，且完成命令的時間相同。

KEYS 的替代方案是 SCAN 命令。它會反覆運算 Keyspace，限制在特定數量的項目中執行反覆運算，避免延長引擎上區塊的執行時間。

掃描具有 COUNT 參數，用來設定反覆運算區塊的大小。預設值為 10 (每個反覆運算 10 個項目)。

根據資料庫中項目數的不同，COUNT 值小的區塊需執行更多的反覆運算才能完成完整掃描，而較大的值每次反覆運算時會讓引擎維持忙碌狀態更久。小計數值會使 SCAN 執行速度較慢，而較大的值可能會對 KEYS 造成上述相同問題。

範例是以計數值 10 執行 SCAN 命令，將需要在具有 1 百萬個索引鍵的資料庫上進行 100,000 次重複作業。如果平均網路封包來回時間為 0.5 毫秒，則傳輸請求會花費大約 50,000 毫秒 (50 秒)。

另一方面，如果計數值為 100,000，則需要執行單次反覆運算，且傳輸只會花 0.5 毫秒。不過，引擎會完全封鎖其他作業，直到該命令完成掃描所有的 Keyspace。

除了 KEYS 以外，如果未正確使用，其他幾個命令可能有害。若要查看所有命令及其各自時間複雜度的清單，請前往 <https://redis.io/commands>。

潛在問題範例：

- **Lua 指令碼**：Redis 提供了一個嵌入式的 Lua 解譯器，允許在伺服器端執行指令碼。Redis 上的 Lua 指令碼會在引擎層級執行，且根據定義為不可部分完成，這表示此指令碼執行時，不允許其他命令或指令碼執行。若使用 Lua 指令碼，便有可能直接在 Redis 引擎執行多個命令、決策演算法、資料剖析以及其他作業。雖然這類指令碼的不可部分完成特性和卸載應用程式的機會很誘人，但將指令碼用於小型作業時必須小心。開啟時 ElastiCache，Lua 指令碼的執行時間限制為 5 秒。未寫入 Keyspace 的指令碼將在 5 秒過後自動終止。為了避免資料損毀和不一致，如果指令碼執行沒有在 5 秒內完成，且執行期間有任何寫入，節點便會容錯移轉。[交易](#)是保證 Redis 中多個相關索引鍵修改具一致性的替代方案。交易允許執行一個命令區塊，觀察現有的索引鍵修改項目。如果觀察的任何索引鍵在交易完成之前有所變更，系統會捨棄所有修改。
- **大量刪除項目**：DEL 命令接受多個參數，這些參數是要刪除的索引鍵名稱。刪除作業會同步進行，如果參數清單很大，或是包含大型清單、集合、排序集合或雜湊 (包含多個子項目的資料結構)，則需花費大量的 CPU 時間。換句話說，如果單一索引鍵有許多元素，即使只是刪除單一索引鍵，也可能需要花相當長的時間。DEL 的替代方案是 UNLINK，這是自 Redis 4 版開始可用的非同步命令。UNLINK 必須盡可能優先於 DEL。從 Redis 6x 開始，`lazyfree-lazy-user-`

`del` 參數會使 DEL 命令的行為類似 ElastiCache 於啟用 UNLINK 時。如需詳細資訊，請參閱 [Redis 6.0 參數變更](#)。

- 對多個索引鍵上產生作用的命令：先前提到的 DEL 是接受多個參數的命令，且其執行時間將直接與此成正比。不過 Redis 還提供許多作用雷同的命令。範例為 MSET 和 MGET 允許一次插入或檢索多個字串索引鍵。使用它們可能有助於減少許多個別 SET 或 GET 命令的固有網路延遲。不過，廣泛的參數清單會影響 CPU 使用率。

雖然單獨 CPU 使用率並不是連線問題的原因，但花太多時間對多個索引鍵處理單一或幾個命令，可能會造成其他請求失敗，並增加整體 CPU 使用率。

索引鍵的數量和大小會影響命令的複雜性，進而影響完成時間。

可以作用於多個索引鍵的其他命令範例

有：HMGET、HMSET、MSETNX、PFCOUNT、PFMERGE、SDIFF、SDIFFSTORE、SINTER、SINTERSTORE 或 ZINTERSTORE。

- 作用於多個資料類型的命令：Redis 也提供作用於一或多個索引鍵的命令 (所有資料類型均適用)。ElastiCache 為 Redis 提供了指標 `KeyBasedCmds` 來監視這些命令。此指標會加總在所選期間內執行下列命令的次數：
  - $O(N)$  複雜性：
    - KEYS
  - $O(1)$ 
    - EXISTS
    - OBJECT
    - PTTL
    - RANDOMKEY
    - TTL
    - TYPE
    - EXPIRE
    - EXPIREAT
    - MOVE
    - PERSIST
    - PEXPIRE
    - PEXPIREAT

- UNLINK ( $O(N)$ ) 用於回收記憶體。但是，記憶體回收任務發生在單獨一個執行序中，且不會阻止引擎
- 根據資料類型而不同的複雜性時間：
  - DEL
  - DUMP
  - RENAME 視為具有  $O(1)$  複雜性的命令，但在 DEL 內部執行。執行時間會依重新命名索引鍵的大小而變動。
  - RENAMENX
  - RESTORE
  - SORT
- 大雜湊：雜湊是一種資料類型，允許單個索引鍵具有多個鍵值子項目。每個雜湊都可以存放 4.294.967.295 個項目，且大雜湊上的作業可能會變得非常昂貴。與 KEYS 類似，雜湊具有帶有  $O(N)$  時間複雜度的 HKEYS 命令， $N$  是雜湊中的項目數。HSCAN 必須優先於 HKEYS 以避免長時間執行命令。HDEL、HGETALL、HMGET、HMSET 和 HVALS 命令應在大雜湊上謹慎使用。
- 其他大數據結構：除了雜湊之外，其他資料結構可能是 CPU 密集型。集合、清單、排序集合和 Hyperloglog 也可能需要花大量時間來操作，取決於其大小和使用的命令。如需這些命令的詳細資訊，請參閱 <https://redis.io/commands>。

## 網路連線能力驗證

檢閱與 DNS 解析、安全群組、網路 ACL 和路由表相關的網路組態之後，可以使用 VPC Reachability Analyzer 和系統工具來驗證連線。

Reachability Analyzer 會測試網路連線能力，並確認是否滿足所有的需求和許可。對於以下測試，您需要 VPC 中其中一個可用 ElastiCache 節點的 ENI ID ( 彈性網路介面識別 )。若要取得此資料，請執行下列操作：

1. 前往 <https://console.aws.amazon.com/ec2/v2/home?#NIC:>
2. 依 ElastiCache 叢集名稱或先前從 DNS 驗證取得的 IP 位址篩選介面清單。
3. 寫下或以其他方式儲存 ENI ID。如果顯示多個介面，請檢閱說明以確認它們屬於正確的 ElastiCache 叢集，然後選擇其中一個介面。
4. 繼續下一個步驟。

5. 在 <https://console.aws.amazon.com/vpc/home?#> 建立分析路徑，ReachabilityAnalyzer然後選擇下列選項：
  - 來源類型：如果您的用 ElastiCache 戶端在 Amazon EC2 執行個體上執行，請選擇執行個體，如果它使用其他服務 (例如具有 aws-vpc 網路的 AWS Fargate Amazon ECS 等)，以及相應的資源 ID (EC2 執行個體或 ENI ID)；AWS Lambda
  - Destination Type (目的地類型)：選擇 Network Interface (網路介面)，然後從清單中選取 ElastiCache ENI。
  - 目標連接埠：為 Redis 指定 6379，或 ElastiCache 針對記憶體快取指定 11211。ElastiCache 這些是使用預設組態定義的連接埠，此範例假設未加以變更。
  - Protocol (通訊協定)：TCP

建立分析路徑，並等待一段時間產生結果。如果狀態為無法連線，請開啟分析詳細資料並檢閱分析總管，取得遭封鎖請求的詳細資訊。

如果通過可連線性測試，請繼續執行作業系統層級的驗證。

若要驗證 ElastiCache 服務連接埠上的 TCP 連線：在 Amazon Linux 上，Nping 可在套件中使用 nmap，可以測試連接 ElastiCache 埠上的 TCP 連線，以及提供建立連線的網路往返時間。使用此選項可驗證 ElastiCache 叢集的網路連線能力和目前的延遲，如下所示：

```
$ sudo nping --tcp -p 6379 example.xxxxxx.ng.0001.use1.cache.amazonaws.com

Starting Nping 0.6.40 ( http://nmap.org/nping ) at 2020-12-30 16:48 UTC
SENT (0.0495s) TCP ...
(Output suppressed )

Max rtt: 0.937ms | Min rtt: 0.318ms | Avg rtt: 0.449ms
Raw packets sent: 5 (200B) | Rcvd: 5 (220B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

根據預設，nping 每傳送 5 次探查會延遲 1 秒。您可以使用「-c」選項來增加探查數，並使用「--delay」來變更傳送新測試的時間。

如果測試出現 nping 失敗，且通過了 VPC Reachability Analyzer 測試，請您的系統管理員檢閱可能的主機式防火牆規則、非對稱式路由規則，或是作業系統層級任何其他可能的限制。

在 ElastiCache 主控台上，檢查 ElastiCache 叢集詳細資料中是否已啟用傳輸中加密。如果已啟用傳輸中加密，請確認是否可以使用下列命令建立 TLS 工作階段：

```
openssl s_client -connect example.xxxxxx.use1.cache.amazonaws.com:6379
```

如果連線和 TLS 交涉成功，預期會有密集的傳出。檢查最後一行中可用的傳回碼，該值必須是 0 (ok)。如果 openssl 傳回不同的內容，請前往以下頁面查看錯誤原因：<https://www.openssl.org/docs/man1.0.2/man1/verify.html#DIAGNOSTICS>。

如果所有基礎結構和作業系統測試都通過，但您的應用程式仍無法連線到 ElastiCache，請檢查應用程式組態是否符合 ElastiCache 設定。常見的錯誤包括：

- 您的應用程式不支持 ElastiCache 集群模式，並啟用 ElastiCache 了集群模式；
- 您的應用程式不支援 TLS/SSL，且 ElastiCache 已啟用傳輸中加密功能；
- 應用程式支援 TLS/SSL，但沒有正確的組態標記或信任的憑證授權機構；

## 網路相關限制

- 連線數上限：同時連線數具有硬性限制。每個 ElastiCache 節點允許在所有用戶端之間同時連線達 65,000 個。您可以透過上的 CurrConnections 測量結果監視此限制 CloudWatch。不過，用戶端也有自己的傳出連線限制。在 Linux 上，請使用下列命令檢查允許的暫時性連接埠範圍：

```
# sysctl net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 32768 60999
```

在前面的範例中，將允許 28231 從相同來源連線到相同目的地 IP (ElastiCache 節點) 和連接埠。下列指令顯示特定 ElastiCache 節點存在多少連線 (IP 1.2.3.4)：

```
ss --numeric --tcp state connected "dst 1.2.3.4 and dport == 6379" | grep -vE
'^State' | wc -l
```

如果數字太高，您的系統可能會因為嘗試處理連線請求而過載。建議考慮實作適用技術，例如連線集區或持續連線，以便更妥善處理連線。盡可能設定連線集區，將連線數上限限制為幾百個。此外，用於處理逾時或其他連線例外狀況的退避邏輯是明智的做法，可避免發生問題的情況下連線流失。

- 網路流量限制：檢查 [Redis 的下列 CloudWatch 指標](#)，以識別 ElastiCache 節點上可能達到的網路限制：
  - NetworkBandwidthInAllowanceExceeded/NetworkBandwidthOutAllowanceExceeded：因為輸送量超過彙總頻寬限制而形成的網路封包。

請務必注意，寫入主節點的每個位元組都會複寫到 N 個複本 (N 是複本數)。具有小型節點類型、多個複本以及密集寫入型請求的叢集，可能無法處理複寫待處理項目。對於這種情況，最佳實務是擴充規模 (更改節點類型)、水平擴展 (在啟用叢集模式的叢集中增加碎片)，減少複本數或最大限度減少寫入次數。

- `NetworkContrackAllowanceExceeded`：因為超過指派給節點的連線數 (在所有安全群組之間追蹤) 上限而形成的封包。這段期間內，新的連線可能會失敗。
- `NetworkPackets PerSecondAllowanceExceeded`：超過每秒的封包數上限。以高極小型請求率為基礎的工作負載，可能會在達到頻寬上限之前達到此限制。

上述指標是確認節點達到網路限制的理想方式。但是，限制也可以藉由網路指標維持平穩來識別。

如果觀察到很長時間都維持平穩，接著可能會出現複寫延遲、用於快取的位元組增加、可釋出的記憶體減少，高交換和 CPU 使用率。Amazon EC2 執行個體也有網路限制，可透過 [ENA 驅動程式指標](#) 追蹤。搭配增強型網路支援和 ENA 驅動程式 2.2.10 或更新版本的 Linux 執行個體，可以使用下列命令檢閱限制計數器：

```
# ethtool -S eth0 | grep "allowance_exceeded"
```

## CPU 使用率

CPU 使用率測量結果是調查的起點，下列項目有助於縮小可能發生的問題範 ElastiCache 圍：

- `Redis SlowLogs`：ElastiCache 默認配置保留最後 128 個命令，這些命令需要超過 10 毫秒才能完成。慢速命令的歷史記錄會在引擎執行時間內保留，且會在故障或重新啟動的情況下遺失。如果清單達到 128 個項目，將會刪除舊活動，為新的活動騰出空間。慢速事件清單的大小和視為緩慢的執行時間，可以透過 [自訂參數群組](#) 中的 `slowlog-max-len` 和 `slowlog-log-slower-than` 參數加以修改。慢速日誌清單可以透過在引擎上執行 `SLOWLOG GET 128` 來擷取，其中 128 是最後回報的 128 個慢速命令。每個項目都有下列欄位：

```
1) 1) (integer) 1 -----> Sequential ID
   2) (integer) 1609010767 --> Timestamp (Unix epoch time)of the Event
   3) (integer) 4823378 -----> Time in microseconds to complete the command.
   4) 1) "keys" -----> Command
      2) "*" -----> Arguments
   5) "1.2.3.4:57004"-> Source
```

上述事件發生在 12 月 26 日 19:26:07 (UTC)，花了 4.8 秒 (4.823 毫秒) 完成，其原因是從用戶端 1.2.3.4 請求的 KEYS 命令。

在 Linux 上，時間戳記可以使用命令日期轉換：

```
$ date --date='@1609010767'  
Sat Dec 26 19:26:07 UTC 2020
```

使用 Python：

```
>>> from datetime import datetime  
>>> datetime.fromtimestamp(1609010767)  
datetime.datetime(2020, 12, 26, 19, 26, 7)
```

或在窗戶上使用 PowerShell：

```
PS D:\Users\user> [datetimeoffset]::FromUnixTimeSeconds('1609010767')  
DateTime          : 12/26/2020 7:26:07 PM  
UtcDateTime       : 12/26/2020 7:26:07 PM  
LocalDateTime    : 12/26/2020 2:26:07 PM  
Date              : 12/26/2020 12:00:00 AM  
Day               : 26  
DayOfWeek         : Saturday  
DayOfYear        : 361  
Hour              : 19  
Millisecond       : 0  
Minute           : 26  
Month             : 12  
Offset            : 00:00:00Ticks           : 637446075670000000  
UtcTicks         : 637446075670000000  
TimeOfDay        : 19:26:07  
Year              : 2020
```

在短時間內 (一分鐘以內) 出現許多緩慢命令是值得關注的原因。檢閱命令的性質及如何最佳化 (請參閱前面的範例)。如果經常回報具有 O(1) 時間複雜性的命令，請檢查之前提到出現高 CPU 使用率的其他因素。

- 延遲指標: ElastiCache 對於 Redis 提供指 CloudWatch 標來監控不同類別命令的平均延遲。資料點的計算方式是將類別中命令的執行總次數除以期間的總執行時間。請務必了解，延遲指標結果是多個命令的彙總結果。單一命令可能會導致非預期的結果 (例如逾時)，而不會對指標造成重大影響。針對這種情況，慢速事件會是更準確的資訊來源。下列清單包含可用的延遲指標，以及影響這些指標的各別命令。
  - EvalBasedCmdsLatency: 與 Lua 腳本命令相關,eval,evalsha;
  - GeoSpatialBasedCmdsLatency: geodist, geohash, geopos, georadius, georadiusbymember, geoadd;
  - GetTypeCmdsLatency : 讀取命令，無論資料類型為何；
  - HashBasedCmdsLatency: hexists, hget, hgetall, hkeys, hlen, hmget, hvals, hstrlen, hdel, hincrby, hincrbyfloat, hmset, hset, hsetnx;
  - HyperLogLogBasedCmdsLatency: pfselftest, pfcount, pfdebug, pfadd, pfmerge;
  - KeyBasedCmdsLatency : 可以對不同數據類型採取行動的命令 : dumpexistskeysobjectpttl,randomkey,ttl,type,del,,expire,expireat,move,persist,punlink
  - ListBasedCmdsLatency: 林德斯, LLEN, 左範圍, BLPOP, brpop 的, brpop, brpoplpush, 林塞特, lpop, lpush, Lpushx 的, LREM, LSET 的, ltrim, rpop, rpoplpush, rpush, rpushx;
  - PubSubBasedCmdsLatency : 訂閱，發布，發布，訂閱，訂閱，取消訂閱;
  - SetBasedCmdsLatency: scard, sdiff, sinter, sismember, smembers, srandmember, sunion, sadd, sdiffstore, sinterstore, smove, spop, srem, sunionstore;
  - SetTypeCmdsLatency : 編寫命令，無論數據類型如何;
  - SortedSetBasedCmdsLatency: zcard, zcount, zrange, zrangebyscore, zrank, zrevrange, zrevrangebyscore, zrevrank, zscore, zrangebylex, zrevrangebylex, zlexcount, zadd, zincrby, zinterstore, zrem, zremrangebyrank, zremrangebyscore, zunionstore, zremrangebylex, zpopmax, zpopmin, bzpopmin, bzpopmax;
  - StringBasedCmdsLatency: bitcount, get, getbit, getrange, mget, strlen, substr, bitpos, append, bitop, bitfield, decr, decrby, getset, incr, incrby, incrbyfloat, mset, msetnx, psetex, set, setbit, setex, setnx, setrange;
  - StreamBasedCmdsLatency: xrange, xrevrange, xlen, xread, xpending, xinfo, xadd, xgroup, readgroup, xack, xclaim, xdel, xtrim, xsetid;
- Redis 執行時間命令 :

- `info commandstats`：提供自 Redis 引擎啟動以來執行的命令清單、累計執行次數、總執行時間以及每個命令的平均執行時間；
- `client list`：提供目前連線的用戶端和相關資訊的清單，如緩衝區使用率、上次執行的命令等；
- Backup 和複寫：ElastiCache 對於 2.8.22 之前的 Redis 版本，請使用分叉處理程序來建立備份，並處理與複本的完整同步。這種方法可能會在寫入密集型使用案例中產生顯著的記憶體額外負荷。

從 ElastiCache Redis 2.8.22 開始，AWS 引入了無叉式備份和複製方法。新方法可能會延遲寫入以防止失敗。這兩種方法都會造成某些時段的 CPU 使用率較高，導致更長的回應時間，進而引發用戶端在執行期間逾時。請一律檢查是否在備份期間內發生用戶端故障或 `SaveInProgress` 指標為 1 的情況。建議您將備份時段排定為使用率低的時段，盡量減少用戶端發生問題或備份失敗的可能性。

## 從伺服器端終止連線

Redis 組態的 ElastiCache 預設值會保留無限期建立用戶端連線。不過在某些情況下可能需終止連線。例如：

- 用戶端應用程式中的錯誤可能會造成連線遭遺忘，並以閒置狀態保持成立。這就是所謂的「連線流失」，結果會發現 `CurConnections` 指標中已成立的連線數穩定增加。此行為可能會導致用戶端或 ElastiCache 端的飽和度。當無法從用戶端立即修正時，有些系統管理員會在其 ElastiCache 參數群組中設定「逾時」值。逾時是讓閒置連線持續存在的時間 (以秒為單位)。如果用戶端沒有在此期間提交任何請求，一旦連線達到逾時值，Redis 引擎就會終止連線。較小的逾時值可能會導致不必要的連線中斷，屆時用戶端需要妥善處理並重新連線，進而導致延遲。
- 用於存放索引鍵的記憶體與用戶端緩衝區共用。具有大型請求或回應的慢速用戶端，可能需要大量的記憶體來處理其緩衝區。Redis 設定 ElastiCache 的預設值不會限制一般用戶端輸出緩衝區的大小。如果達到 `maxmemory` 上限，引擎將會嘗試移出項目以滿足緩衝區使用量。在記憶體極低的情況下，ElastiCache Redis 可能會選擇中斷耗用大型用戶端輸出緩衝區的用戶端連線，以釋放記憶體並保留叢集的健全狀況。

可以使用自訂組態來限制用戶端緩衝區的大小，且用戶端達到限制時將會中斷連線。不過用戶端應該能處理未預期的中斷連線。處理一般用戶端的緩衝區大小的參數如下：

- `client-query-buffer-limit`：單個輸入請求的最大大小；
- `client-output-buffer-limit-normal-soft-limit`：用戶端連線的軟限制。如果超過軟限制的時間超過上定義的時間 (以秒為單位) `normal-soft-seconds` 或達到硬限制，則連接將終止；`client-output-buffer-limit`
- `client-output-buffer-limit-normal-soft-seconds`：允許的時間超過 `client-output-buffer-limit` 的連接 `normal-soft-limit`;

- `client-output-buffer-limit-normal-hard-limit`：達到此限制的連線會立即終止。

除了一般用戶端緩衝區，下列選項也可控制複本節點和 Pub/Sub (發佈/訂閱) 用戶端的緩衝區：

- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-replica-soft-seconds`;
- `client-output-buffer-limit-replica-hard-limit`;
- `client-output-buffer-limit-pubsub-soft-limit`;
- `client-output-buffer-limit-pubsub-soft-seconds`;
- `client-output-buffer-limit-pubsub-hard-limit`;

## Amazon EC2 執行個體的用戶端疑難排解

客戶端的負載和響應也會影響到的請求 ElastiCache。疑難排解間歇性連線或逾時問題時，需要仔細檢閱 EC2 執行個體和作業系統限制。幾項需觀察的要點：

- CPU：
  - EC2 執行個體 CPU 使用率：確定 CPU 尚未飽和或接近 100%。歷史分析可以通過以下方式完成 CloudWatch，但請記住，數據點粒度可以是 1 分鐘 (啟用詳細監控) 或 5 分鐘；
  - 如果使用 [高載 EC2 執行個體](#)，請確定其 CPU 餘額尚未耗盡。此資訊可用於 CPU Credit Balance CloudWatch 測量結果。
  - 短時間的高 CPU 使用率可能會導致逾時，而不會反映 100% 的使用率。CloudWatch 這種情況需使用作業系統工具 (如 `top`、`ps` 和 `mpstat`) 進行即時監控。
- 網路
  - 根據執行個體容量檢查網路輸送量是否低於可接受的值。如需詳細資訊，請參閱 [Amazon EC2 執行個體類型](#)
  - 在具有 `ena` 增強型網路驅動程式的執行個體上，查看 [ENA 統計數字](#) 以取得逾時或超出的限制。下列統計數字有助於確認網路限制飽和情況：
    - `bw_in_allowance_exceeded/bw_out_allowance_exceeded`：因過量的傳入或傳出輸送量而造成的封包數；
    - `contrack_allowance_exceeded`：由於安全群組 [連線追蹤限制](#) 而捨棄的封包數。當此限制飽和時，新的連線將會失敗；
    - `linklocal_allowance_exceeded`：由於透過 VPC DNS 對執行個體中繼資料 NTP 發出過多請求而捨棄的封包數。所有服務的限制都是每秒 1024 個封包；

- `pps_allowance_exceeded` : 由於每秒封包比率過高而捨棄的封包數。當網路流量包含每秒數千或數百萬個極小的請求時，便可能達到 PPS 限制。ElastiCache 可以優化流量，通過管道或一次執行多個操作的命令來更好地利用網絡數據包，MGET 而不是 GET。

## 解析完成單個請求所花費的時間

- 在網絡上：Tcpdump 和 Wireshark ( 命令行上的 tshark ) 是方便的工具，以了解請求花了多少時間旅行網絡，擊中 ElastiCache 引擎並獲得回報。下列範例說明如何使用下列命令建立單一請求：

```
$ echo ping | nc example.xxxxxx.ng.0001.use1.cache.amazonaws.com 6379
+PONG
```

tcpdump 與上面的命令同時執行並傳回了：

```
$ sudo tcpdump -i any -nn port 6379 -tt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
1609428918.917869 IP 172.31.11.142.40966
    > 172.31.11.247.6379: Flags [S], seq 177032944, win 26883, options [mss
    8961,sackOK,TS val 27819440 ecr 0,nop,wscale 7], length 0
1609428918.918071 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [S.], seq
    53962565, ack 177032945, win
    28960, options [mss 1460,sackOK,TS val 3788576332 ecr 27819440,nop,wscale 7],
    length 0
1609428918.918091 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.] , ack 1, win
    211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918122
    IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [P.], seq 1:6, ack 1, win 211,
    options [nop,nop,TS val 27819440 ecr 3788576332], length 5: RESP "ping"
1609428918.918132 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [F.], seq 6, ack
    1, win 211, options [nop,nop,TS val 27819440 ecr 3788576332], length 0
1609428918.918240 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [.] , ack 6, win
    227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918295
    IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [P.], seq 1:8, ack 7, win 227,
    options [nop,nop,TS val 3788576332 ecr 27819440], length 7: RESP "PONG"
1609428918.918300 IP 172.31.11.142.40966 > 172.31.11.247.6379: Flags [.] , ack 8, win
    211, options [nop,nop,TS val 27819441 ecr 3788576332], length 0
1609428918.918302 IP 172.31.11.247.6379 > 172.31.11.142.40966: Flags [F.], seq 8, ack
    7, win 227, options [nop,nop,TS val 3788576332 ecr 27819440], length 0
1609428918.918307
```



```
1609430221.747548 (+ 0.001887) recvfrom(0, 0x7ffcf2f2ca50, 8192,
    0, 0x7ffcf2f2c9d0, [128]) = -1 ENOTSOCK (Socket operation on non-socket)
1609430221.747858 (+ 0.000308) sendto(3, "ping\n", 5, 0, NULL, 0) = 5
1609430221.748048 (+ 0.000188) recvfrom(0, 0x7ffcf2f2ca50, 8192, 0, 0x7ffcf2f2c9d0,
    [128]) = -1 ENOTSOCK
    (Socket operation on non-socket)
1609430221.748330 (+ 0.000282) recvfrom(3, "+PONG\r\n", 8192, 0, 0x7ffcf2f2c9d0,
    [128->0]) = 7
+PONG
1609430221.748543 (+ 0.000213) recvfrom(3, "", 8192, 0, 0x7ffcf2f2c9d0, [128->0]) = 0
1609430221.752110
    (+ 0.003569) +++ exited with 0 +++
```

在上面的例子中，該命令花費超過 54 毫秒才完成 (752110 - 697712 = 54398 微秒)。

將 nc 執行個體化並執行名稱解析花了約 20 毫秒的大量時間 (從 697712 到 717890)，在這之後，花了 2 毫秒來建立 TCP 通訊端 (745659 到 747858)，再花 0.4 毫秒 (747858 到 748330) 提交和接收請求的回應。

# Amazon ElastiCache 的安全性

雲端安全是 AWS 最重視的一環。身為 AWS 客戶的您，將能從資料中心和網路架構的建置中獲益，以滿足組織最為敏感的安全要求。

安全是 AWS 與您共同的責任。[共同的責任模型](#) 將此描述為雲端 本身 的安全和雲端內部的安全：

- 雲端本身的安全 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。在 [AWS 合規計畫](#) 中，第三方稽核員會定期測試並驗證我們的安全功效。若要了解適用於 Amazon ElastiCache 的合規計畫，請參閱 [合規計畫範圍內的 AWS 服務](#)。
- 雲端內部的安全 – 您的責任取決於所使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您公司的請求和適用法律和法規。

本文件有助於您了解如何在使用 Amazon ElastiCache 時套用共同的責任模型。下列主題說明如何將 Amazon ElastiCache 設定為符合您的安全與合規目標。您也將了解如何使用其他 AWS 服務，幫助您監控並保護 Amazon ElastiCache 資源。

## 主題

- [Amazon ElastiCache 中的資料保護](#)
- [網際網路流量隱私權](#)
- [Amazon Identity and Access Management ElastiCache](#)
- [Amazon 的合規驗證 ElastiCache](#)
- [Amazon ElastiCache 的備援功能](#)
- [AWS ElastiCache 中的基礎設施安全性](#)
- [服務更新 ElastiCache](#)

## Amazon ElastiCache 中的資料保護

AWS [共同的責任模型](#) 適用於 AWS ElastiCache (ElastiCache) 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端的全球基礎設施。您必須負責維護在此基礎設施上託管之內容的控制權。此內容包括您所使用 AWS 服務的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱 [資料隱私權常見問答集](#)。

為了保護資料，我們建議您保護 AWS 帳戶登入資料，並使用 AWS Identity and Access Management (IAM) 來設定個別帳戶。如此一來，每個使用者都只會獲得授予完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 TLS 與 AWS 資源進行通訊。
- 使用 AWS CloudTrail 設定 API 和使用者活動記錄。
- 使用 AWS 加密解決方案，以及 AWS 服務內的所有預設安全控制。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Simple Storage Service (Amazon Simple Storage Service (Amazon S3)) 的個人資料。

我們強烈建議您絕對不要將客戶帳戶號碼等敏感的識別資訊，放在自由格式的欄位中，例如 Name (名稱) 欄位。這包括使用 ElastiCache 或使用主控台、API、AWS CLI 或 AWS SDK 的其他 AWS 服務時。您在 ElastiCache 或其他服務中輸入的任何資料都可能選入診斷日誌中。當您提供外部伺服器的 URL 時，請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

## 主題

- [Amazon ElastiCache 的資料安全性](#)

## Amazon ElastiCache 的資料安全性

為了協助保護您的資料安全，Amazon ElastiCache 和 Amazon EC2 提供保護您伺服器上的資料不受未授權存取的機制。

Amazon ElastiCache for Memcached 也為執行 Memcached 1.6.12 或更新版本的快取上的資料提供了加密功能。

- 傳輸中加密會在您的資料從某一處移到另一處時進行加密，例如在您叢集內的節點之間移動，或是在您的快取與應用程式之間移動。
- 待用加密可在同步和備份操作期間加密您磁碟上的資料。

## 主題

- [ElastiCache 傳輸中加密 \(TLS\)](#)
- [ElastiCache 中的靜態加密](#)

## ElastiCache 傳輸中加密 (TLS)

為了協助確保資料安全，Amazon ElastiCache 和 Amazon EC2 提供機制以防止未經授權存取伺服器上的資料。透過提供傳輸中加密功能，ElastiCache 您可以使用工具在資料從一個位置移至另一個位置時協助保護資料。

所有無伺服器快取都會啟用傳輸中加密。對於自行設計的叢集，您可以在使用 `CreateCacheCluster` (CLI : `create-cache-cluster`) 操作建立快取叢集時，將參數 `TransitEncryptionEnabled` 設定為 `true` (CLI : `--transit-encryption-enabled`)，以在快取叢集上啟用傳輸中加密。

### 主題

- [傳輸中加密概觀](#)
- [傳輸中加密條件](#)
- [傳輸中加密最佳實務](#)
- [啟用傳輸中加密](#)
- [使用 OpenSSL 連接至啟用傳輸中加密的節點](#)
- [使用 Java 建立 TLS Memcached 用戶端](#)
- [使用 PHP 建立 TLS Memcached 用戶端](#)

### 傳輸中加密概觀

Amazon ElastiCache 傳輸中加密是一項功能，可讓您在資料最容易受到攻擊的位置 (當資料從一個位置傳輸到另一個位置時) 提高資料的安全性。因為要加密和解密端點的資料需要一些處理，啟用傳輸中加密可能會有一些效能影響。您應該對您具有和不具有傳輸中加密的資料進行基準分析，以判斷對您的使用案例的影響。

ElastiCache 傳輸中加密實作下列功能：

- 加密的用戶端連線：用戶端與快取節點的連線會經過 TLS 加密。
- 加密的伺服器連線：在叢集中的節點之間移動的資料會經過加密。
- 伺服器身分驗證 - 用戶端可以驗證是否已連線至正確的伺服器。

### 傳輸中加密條件

規劃自行設計的叢集 ElastiCache 實作時，應牢記下列 Amazon 傳輸中加密的限制：

- 執行 Memcached 1.6.12 和更新版本的叢集上支援傳輸中加密。
- 傳輸中加密支援 Transport Layer Security (TLS) 1.2 和 1.3 版。
- 只有在 Amazon VPC 中執行的叢集支援傳輸中加密。
- 執行下列節點類型的複寫群組不支援傳輸中加密：M1、M2、M3、R3、T2。

如需詳細資訊，請參閱 [支援的節點類型](#)。

- 透過將參數 `TransitEncryptionEnabled` 明確設定為 `true`，可啟用傳輸中加密。
- 您只能在建立叢集時，在叢集上啟用傳輸中加密。您無法透過修改叢集來開啟和關閉傳輸中加密。
- 請確定您的快取用戶端支援 TLS 連線，而且您已在用戶端組態中啟用它。

### 傳輸中加密最佳實務

- 因為要加密和解密端點的資料需要一些處理，實作傳輸中加密可能會降低效能。對您自己的資料進行傳輸中加密與無加密基準分析的比較，來判斷對您實作的影響。
- 因為建立新連線可能所費不貲，您可以透過持續保存您的 TLS 連線來減少傳輸中加密的效能影響。

### 啟用傳輸中加密

若要在使用 AWS 管理主控台建立 Memcached 叢集時，啟用傳輸中加密，請選取以下項目：

- 選擇 Memcached 當作您的引擎。
- 選擇引擎 1.6.12 或更新版本。
- 在 Encryption in transit (傳輸中加密) 中，選擇 Enable (啟用)。

有關該 step-by-step 過程，請參閱 [創建 Memcached 集群 \(控制台\)](#)。

### 使用 Openssl 連接至啟用傳輸中加密的節點

若要從啟 ElastiCache 用傳輸中加密的 Memcached 節點存取資料，您需要使用搭配安全通訊端層 (SSL) 運作的用戶端。您也可以在 Amazon Linux 和 Amazon Linux 2 上使用 `Openssl s_client`。

在 Amazon Linux 或 Amazon Linux 2 上，使用 `Openssl s_client` 來連線到啟用傳輸中加密的 Memcached 叢集：

```
/usr/bin/openssl s_client -connect memcached-node-endpoint:memcached-port
```

## 使用 Java 建立 TLS Memcached 用戶端

若要在 TLS 模式下建立用戶端，請執行下列操作，使用適當的 SSLContext 來初始化用戶端：

```
import java.security.KeyStore;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManagerFactory;
import net.spy.memcached.AddrUtil;
import net.spy.memcached.ConnectionFactoryBuilder;
import net.spy.memcached.MemcachedClient;
public class TLSDemo {
    public static void main(String[] args) throws Exception {
        ConnectionFactoryBuilder connectionFactoryBuilder = new
ConnectionFactoryBuilder();
        // Build SSLContext
        TrustManagerFactory tmf =
TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
        tmf.init((KeyStore) null);
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, tmf.getTrustManagers(), null);
        // Create the client in TLS mode
        connectionFactoryBuilder.setSSLContext(sslContext);
        MemcachedClient client = new MemcachedClient(connectionFactoryBuilder.build(),
AddrUtil.getAddresses("mycluster.fnjyzo.cfg.use1.cache.amazonaws.com:11211"));

        // Store a data item for an hour.
        client.set("theKey", 3600, "This is the data value");
    }
}
```

## 使用 PHP 建立 TLS Memcached 用戶端

若要在 TLS 模式下建立用戶端，請執行下列操作，使用適當的 SSLContext 來初始化用戶端：

```
<?php

/**
 * Sample PHP code to show how to create a TLS Memcached client. In this example we
 * will use the Amazon ElastiCache Auto Discovery feature, but TLS can also be
 * used with a Static mode client.
 * See Using the ElastiCache Cluster Client for PHP (https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/AutoDiscovery.Using.ModifyApp.PHP.html) for more
 * information
 */
```

```
* about Auto Discovery and persistent-id.
*/

/* Configuration endpoint to use to initialize memcached client.
 * this is only an example */
$server_endpoint = "mycluster.fnjyzo.cfg.use1.cache.amazonaws.com";

/* Port for connecting to the cluster.
 * This is only an example */
$server_port = 11211;

/* Initialize a persistent Memcached client and configure it with the Dynamic client
mode */
$tls_client = new Memcached('persistent-id');
$tls_client->setOption(Memcached::OPT_CLIENT_MODE, Memcached::DYNAMIC_CLIENT_MODE);

/* Add the memcached's cluster server/s */
$tls_client->addServer($server_endpoint, $server_port);

/* Configure the client to use TLS */
if(!$tls_client->setOption(Memcached::OPT_USE_TLS, 1)) {
    echo $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* Set your TLS context configurations values.
 * See MemcachedTLSContextConfig in memcached-api.php for all configurations */
$tls_config = new MemcachedTLSContextConfig();
$tls_config->hostname = '*.mycluster.fnjyzo.use1.cache.amazonaws.com';
$tls_config->skip_cert_verify = false;
$tls_config->skip_hostname_verify = false;

/* Use the created TLS context configuration object to create OpenSSL's SSL_CTX and set
it to your client.
 * Note: These TLS context configurations will be applied to all the servers connected
to this client. */
$tls_client->createAndSetTLSContext((array)$tls_config);

/* test the TLS connection with set-get scenario: */

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

```
    print "Successfully stored key\n";
} else {
    echo "Failed to set key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}

/* retrieve the key */
if ($tls_client->get('key') === 'value') {
    print "Successfully retrieved key\n";
} else {
    echo "Failed to get key: ", $tls_client->getLastErrorMessage(), "\n";
    exit(1);
}
```

如需使用 PHP 用戶端的詳細資訊，請參閱[安裝適用於 PHP 的 ElastiCache 叢集用戶端](#)。

## ElastiCache 中的靜態加密

為了協助保護您的資料安全，Amazon ElastiCache 和 Amazon S3 提供不同方式，可用於限制存取位於快取中資料。如需詳細資訊，請參閱[Amazon VPC 和 ElastiCache 安全性](#)及[Amazon Identity and Access Management ElastiCache](#)。

- 同步和交換操作期間的磁碟

ElastiCache 提供靜態的預設 (服務管理的) 加密，也能讓您在 [AWS Key Management Service \(KMS\)](#) 中使用自己的對稱式客戶受管 AWS KMS 金鑰。備份快取後，在加密選項中，選擇要使用預設加密金鑰或客戶受管金鑰。如需更多詳細資訊，請參閱 [啟用靜態加密](#)。

### Note

預設 (受管服務) 加密是 GovCloud (US) 區域中唯一的可用選項。

只有在快取上建立複寫群組後，才能啟用靜態加密。因為要加密和解密資料需要經過一些處理，因此在這些操作期間啟用靜態加密可能會影響效能。您應該針對使用靜態加密和未使用靜態加密的資料進行基準分析，以判斷對使用案例的效能影響。

### 主題

- [靜態加密條件](#)
- [使用 AWS KMS 中的客戶受管金鑰](#)
- [啟用靜態加密](#)
- [另請參閱](#)

### 靜態加密條件

規劃 ElastiCache 靜態加密實作時，應謹記下列 ElastiCache 靜態加密的限制：

- 只有無伺服器快取支援靜態加密。
- AWS GovCloud (us-gov-east-1 和 us-gov-west-1) 區域不提供使用客戶受管金鑰進行靜態加密的選項。

## 使用 AWS KMS 中的客戶受管金鑰

ElastiCache 支援使用對稱式客戶受管 AWS KMS 金鑰 (KMS 金鑰) 進行靜態加密。客戶受管 KMS 金鑰是您在 AWS 帳戶中建立、擁有和管理的加密金鑰。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [AWS KMS 金鑰](#)。必須先在 AWS KMS 中建立金鑰，才能搭配 ElastiCache 使用。

若要了解如何建立 AWS KMS 根金鑰，請參閱 AWS Key Management Service 開發人員指南中的 [建立金鑰](#)。

ElastiCache 能與 AWS KMS 整合。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [使用授權](#)。無需客戶動作即可啟用 Amazon ElastiCache 與 AWS KMS 的整合。

kms:ViaService 條件索引鍵會將 AWS KMS 金鑰 (KMS 金鑰) 的使用限制為來自指定 AWS 服務的請求。若要搭配 ElastiCache 使用 kms:ViaService，請在條件索引鍵值中包含兩個 ViaService 名稱：elasticache.AWS\_region.amazonaws.com 和 dax.AWS\_region.amazonaws.com。如需詳細資訊，請參閱 [kms:ViaService](#)。

您可以使用 [AWS CloudTrail](#) 來追蹤 Amazon ElastiCache 代表您傳送給 AWS Key Management Service 的請求。向與客戶受管金鑰相關的 AWS Key Management Service 發出的所有 API 呼叫，都具有對應的 CloudTrail 日誌。您也可以透過呼叫 [ListGrants](#) KMS API 呼叫來查看 ElastiCache 建立的授與。

- 如果刪除金鑰或**停用**金鑰，並針對您用來加密快取的金鑰**撤銷授權**，則快取將變成無法恢復。換句話說，在硬體故障後將無法修改或復原。AWSKMS 至少等待七天後，才會刪除根金鑰。金鑰刪除後，您可以使用其他客戶受管金鑰建立備份以用於封存。
- 自動金鑰輪換會保留 AWS KMS 根金鑰的屬性，因此輪換並不影響您存取 ElastiCache 資料的能力。加密的 Amazon ElastiCache 快取不支援手動金鑰輪換，其中包含建立新的根金鑰和更新任何舊金鑰的參照。如需詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [輪換 AWS KMS 金鑰](#)。
- 每個快取都需有一個授權，才能使用 KMS 金鑰加密 ElastiCache 快取。此授權將在快取的整個生命週期中使用。
- 如需 AWS KMS 授權和限制的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的 [限制](#)。

## 啟用靜態加密

所有無伺服器快取都會啟用靜態加密。

您可在建立 ElastiCache 快取時啟用靜態加密。您可以使用 AWS Management Console、AWS CLI 或 ElastiCache API 來執行此作業。

建立快取時，您可以選擇下列其中一個選項：

- 預設 – 此選項使用服務管理的靜態加密。
- 客戶受管金鑰 - 此選項允許您從 AWS KMS 提供金鑰 ID/ARN 以用於靜態加密。

若要了解如何建立 AWS KMS 根金鑰，請參閱 AWS Key Management Service 開發人員指南中的[建立金鑰](#)

內容

- [使用 AWS Management Console 啟用靜態加密](#)

使用 AWS Management Console 啟用靜態加密

在無伺服器快取上啟用靜態加密 (主控台)

所有無伺服器快取都會啟用靜態加密。預設是使用 AWS 擁有的 KMS 金鑰來加密資源。若要選擇您自己的 AWS KMS 金鑰，請進行下列選擇：

- 展開預設設定區段。
- 在預設設定區段下選擇自訂預設設定。
- 在安全性區段下選擇自訂安全性設定。
- 在加密金鑰設定下選擇客戶受管 CMK。
- 在 AWS KMS 金鑰設定下選擇一個金鑰。

另請參閱

- [Amazon VPC 和 ElastiCache 安全性](#)
- [Amazon Identity and Access Management ElastiCache](#)

## 網際網路流量隱私權

Amazon ElastiCache 使用下列技術保護您的快取資料，並資料它不受未經授權的存取：

- [Amazon VPC 和 ElastiCache 安全性](#) 說明安裝所需的安全群組類型。

- [Amazon Identity and Access Management ElastiCache](#) 用於授予和限制使用者、群組和角色的動作。

## Amazon VPC 和 ElastiCache 安全性

因為資料安全很重要，ElastiCache 可用來控制可存取您資料的使用者。您控制資料存取的方式，取決於您是在 Amazon Virtual Private Cloud (Amazon VPC) 還是 Amazon EC2-Classical 中啟動叢集。

### Important

我們已淘汰將 Amazon EC2-Classical 用於啟動 ElastiCache 叢集的做法。所有目前世代的節點只會在 Amazon Virtual Private Cloud 中啟動。

Amazon Virtual Private Cloud (Amazon VPC) 服務會定義虛擬網路，與傳統資料中心幾乎一模一樣。設定 Amazon VPC 時，您可以選擇其 IP 地址範圍、建立子網路，以及設定路由表、網路閘道與安全設定。您也可以將快取叢集新增至虛擬網路，然後使用 Amazon VPC 安全群組來控制快取叢集的存取權。

本節說明如何在 Amazon VPC 中手動設定 ElastiCache 叢集。本資訊的預期對象是想要更深入了解 ElastiCache 和 Amazon VPC 如何搭配運作的使用者。

### 主題

- [了解 ElastiCache 和 Amazon VPC](#)
- [在 Amazon VPC 中存取 ElastiCache 快取的存取模式](#)
- [建立 Virtual Private Cloud \(VPC\)](#)
- [連線到 Amazon VPC 中執行的快取](#)

## 了解 ElastiCache 和 Amazon VPC

ElastiCache 與 Amazon Virtual Private Cloud (Amazon VPC) 完全整合。對 ElastiCache 使用者來說，這代表了下列意義：

- 如果 AWS 帳戶僅支援 EC2-VPC 平台，ElastiCache 一律會在 Amazon VPC 中啟動叢集。
- 如果您是 AWS 的新使用者，系統會將叢集部署至 Amazon VPC。系統將為您建立預設的 VPC。
- 如果您有預設的 VPC，且在啟動叢集時未指定子網路，叢集會於預設 Amazon VPC 啟動。

如需詳細資訊，請參閱[偵測支援的平台以及您是否有預設 VPC](#)。

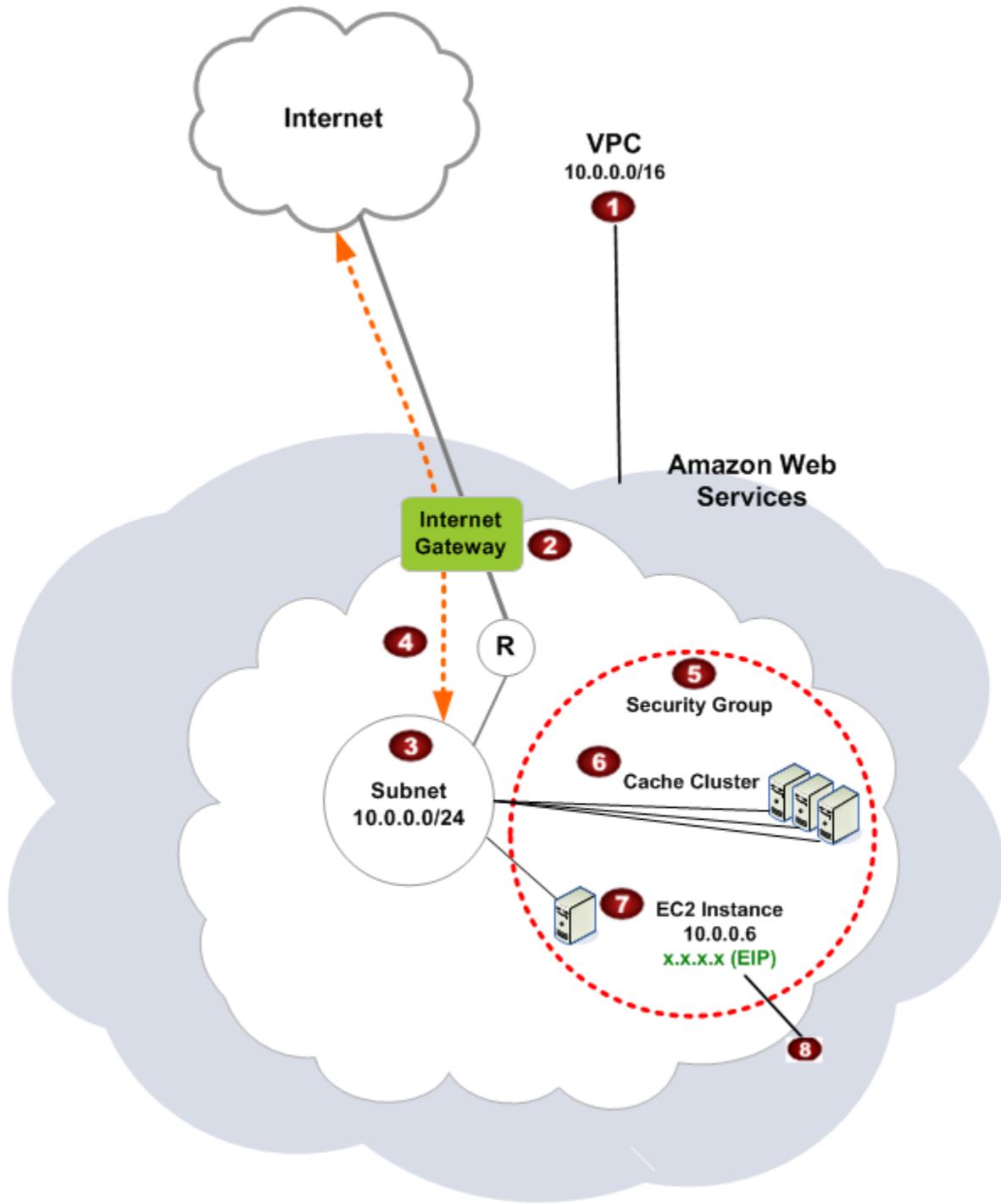
使用 Amazon Virtual Private Cloud，您可以在 AWS 雲端中建立與傳統資料中心幾乎一模一樣的虛擬網路。您可以設定 Amazon VPC，包括選取其 IP 地址範圍、建立子網路和設定路由表、網路閘道和安全設定。

ElastiCache 的基本功能，與在 Virtual Private Cloud 中相同；不論您的叢集是部署在 Amazon VPC 的內部或外部，ElastiCache 都會管理軟體升級、修補、故障偵測和復原。

在 Amazon VPC 外部部署的 ElastiCache 快取節點會獲派一個 IP 地址，為端點/DNS 名稱解析所得出。這樣便能從 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體進行連線。在 Amazon VPC 私有子網路中啟動 ElastiCache 叢集時，每個快取節點會獲指派該子網路內的私有 IP 地址。

### Amazon VPC 中的 ElastiCache 概觀

下圖和下表描述 Amazon VPC 環境，以及在 Amazon VPC 中啟動的 ElastiCache 叢集和 Amazon EC2 執行個體。



1

Amazon VPC 是 AWS 雲端的隔離部分，該雲端獲派自己的 IP 地址區塊。

2

網際網路閘道會將 Amazon VPC 直接連接至網際網路，並提供其他 AWS 資源的存取權，例如在 Amazon VPC 外部執行的 Amazon Simple Storage Service (Amazon S3)。

3

Amazon VPC 子網路是 Amazon VPC IP 地址範圍的區段，您可以在其中根據安全和操作需求隔離 AWS 資源。

4

Amazon VPC 中的路由表會在子網路與網際網路之間導向網路流量。Amazon VPC 有隱含的路由器，在此圖表中以內含 R 的圓形標示。

5

Amazon VPC 安全群組會控制 ElastiCache 叢集和 Amazon EC2 執行個體的傳入和傳出流量。

6

您可以在子網路中啟動 ElastiCache 叢集。快取節點具有來自子網路地址範圍的私有 IP 地址。

7

您也可以在此子網路中啟動 Amazon EC2 執行個體。每個 Amazon EC2 執行個體具有來自子網路之地址範圍的私有 IP 地址。Amazon EC2 執行個體可以連線至相同子網路中的任何快取節點。

8

若要讓 Amazon VPC 中的 Amazon EC2 執行個體可透過網際網路連接，您必須為執行個體指派一個靜態的公有地址 (稱為彈性 IP 地址)。

## 必要條件

若要在 Amazon VPC 中建立 ElastiCache 叢集，您的 Amazon VPC 必須符合下列要求：

- Amazon VPC 必須允許非專用的 Amazon EC2 執行個體。您無法在設定為使用專用執行個體租用戶的 Amazon VPC 中使用 ElastiCache。
- 必須為您的 Amazon VPC 定義快取子網路群組。ElastiCache 會使用該快取子網路群組選取子網路及該子網路內的 IP 地址，以與您的 VPC 端點或快取節點建立關聯。
- 每個子網路的 CIDR 區塊必須夠大，才能為 ElastiCache 提供備用 IP 地址，以在維護活動期間使用。

## 路由和安全性

您可以在 Amazon VPC 中設定路由，以控制流量流動的位置 (例如，通往網際網路閘道或虛擬私有閘道)。利用網際網路閘道，Amazon VPC 可以直接存取不在 Amazon VPC 中執行的其他 AWS 資源。如果您選擇只有一個虛擬私有閘道具有您的組織區域網路的連線，您可以透過 VPN 路由網際網路入站流量，並使用本機安全政策和防火牆來控制傳入。在該情況下，透過網際網路存取 AWS 資源時，您會衍生額外的頻寬費用。

您可以使用 Amazon VPC 安全群組來協助保護 Amazon VPC 中的 ElastiCache 叢集和 Amazon EC2 執行個體。安全群組會在執行個體層級 (而非子網路層級) 以防火牆形式運作。

### Note

強烈建議您使用 DNS 名稱來連線至快取節點，因為基礎 IP 地址可能變更。

## Amazon VPC 文件

Amazon VPC 有專屬的一套文件，說明如何建立和使用 Amazon VPC。下表提供 Amazon VPC 指南的連結。

描述	文件
如何開始使用 Amazon VPC	<a href="#">Amazon VPC 入門</a>
透過 AWS Management Console 使用 Amazon VPC	《 <a href="#">Amazon VPC 使用者指南</a> 》
所有 Amazon VPC 命令的完整描述	<a href="#">Amazon EC2 命令列參考</a> (Amazon VPC 命令均列在 Amazon EC2 參考中)
Amazon VPC API 作業、資料類型和錯誤的完整描述	<a href="#">Amazon EC2 API 參考</a> (Amazon VPC API 作業均列在 Amazon EC2 參考中)
需要在選用的 IPsec VPN 連線中的您那一端設定閘道之網路管理員的資訊	<a href="#">什麼是 AWS Site-to-Site VPN ?</a>

如需更多 Amazon Virtual Private Cloud 的詳細資訊，請參閱 [Amazon Virtual Private Cloud](#)。

## 在 Amazon VPC 中存取 ElastiCache 快取的存取模式

Amazon ElastiCache 支持以下情況來訪問 Amazon VPC 中的緩存：

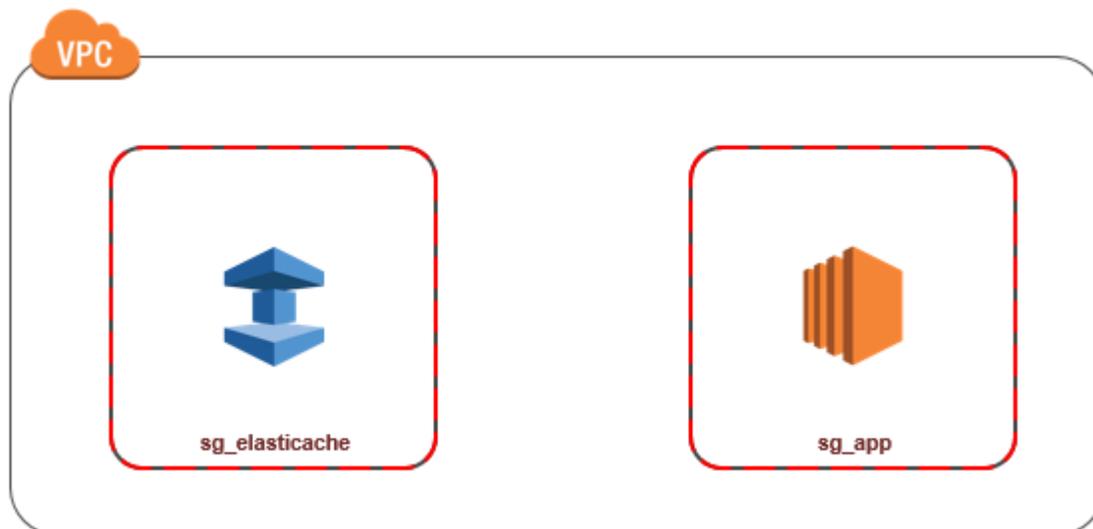
### 內容

- [當它和 Amazon EC2 實例位於同一個 Amazon VPC 時訪問 ElastiCache 緩存](#)
- [當它和 Amazon EC2 實例位於不同的 Amazon VPC 時訪問 ElastiCache 緩存](#)
  - [在 ElastiCache 快取和 Amazon EC2 執行個體位於同一區域的不同 Amazon VPC 時存取快取](#)
    - [使用 Transit Gateway](#)
  - [在 ElastiCache 快取和 Amazon EC2 執行個體位於不同區域的不同 Amazon VPC 時存取快取](#)
    - [使用傳輸 VPC](#)
- [從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
  - [使用 VPN 連線能力，從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
  - [使用直 Connect 從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)

當它和 Amazon EC2 實例位於同一個 Amazon VPC 時訪問 ElastiCache 緩存

最常見的使用案例是部署於 EC2 執行個體的應用程式時，需要連線至相同 VPC 中的快取。

下圖說明此案例。



若要管理相同 VPC 中 EC2 執行個體與快取之間的存取權，最簡單的方式如下：

1. 為快取建立 VPC 安全群組。此安全群組可用來限制快取的存取權。舉例來說，您可以為此安全群組建立自訂規則，以允許使用您建立快取時為其指派的連接埠，以及您將用來存取快取的 IP 地址進行 TCP 存取。

Memcached 快取的預設連接埠是 11211。

2. 為您的 EC2 執行個體建立 VPC 安全群組 (Web 和應用程式伺服器)。若有需要，此安全群組可允許透過 VPC 路由表存取網際網路上的 EC2 執行個體。舉例來說，您可以在此安全群組上設定規則，允許 TCP 透過連接埠 22 存取 EC2 執行個體。
3. 在安全群組中為快取建立自訂規則，以允許來自您為 EC2 執行個體所建立之安全群組的連線。這樣做會允許安全群組的所有成員存取快取。

#### Note

如果您打算使用 [本地區域](#)，請確認本地區域已啟用。在本地區域中建立子網路群組時，VPC 會延伸至該本地區域，VPC 會將子網路視為任何其他可用區域中的任何子網路群組。所有相關閘道和路由表都將自動調整。

在允許來自其他安全群組連線的 VPC 安全群組中建立規則

1. 登入 AWS 管理主控台，然後開啟 Amazon VPC 主控台，網址為 <https://console.aws.amazon.com/vpc>。
2. 在導覽窗格中，選擇安全群組。
3. 選取或建立您將用於快取的安全群組。在 Inbound Rules (傳入規則) 下方，選取 Edit Inbound Rules (編輯傳入規則)，然後選取 Add Rule (新增規則)。此安全群組將允許其他安全群組成員存取。
4. 從 Type (類型) 選擇 Custom TCP Rule (自訂 TCP 規則)。
  - a. 針對連接埠範圍，指定您在建立快取時所使用的連接埠。

Memcached 快取的預設連接埠是 11211。

- b. 在 Source (來源) 方塊中輸入安全群組的 ID。從清單中選取您將用於 Amazon EC2 執行個體的安全群組。
5. 完成後，請選擇 Save (儲存)。

### Edit inbound rules

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	6379	Custom sg_app

Add Rule

sg-99fc5ce6 - sg\_app

Cancel Save

當它和 Amazon EC2 實例位於不同的 Amazon VPC 時訪問 ElastiCache 緩存

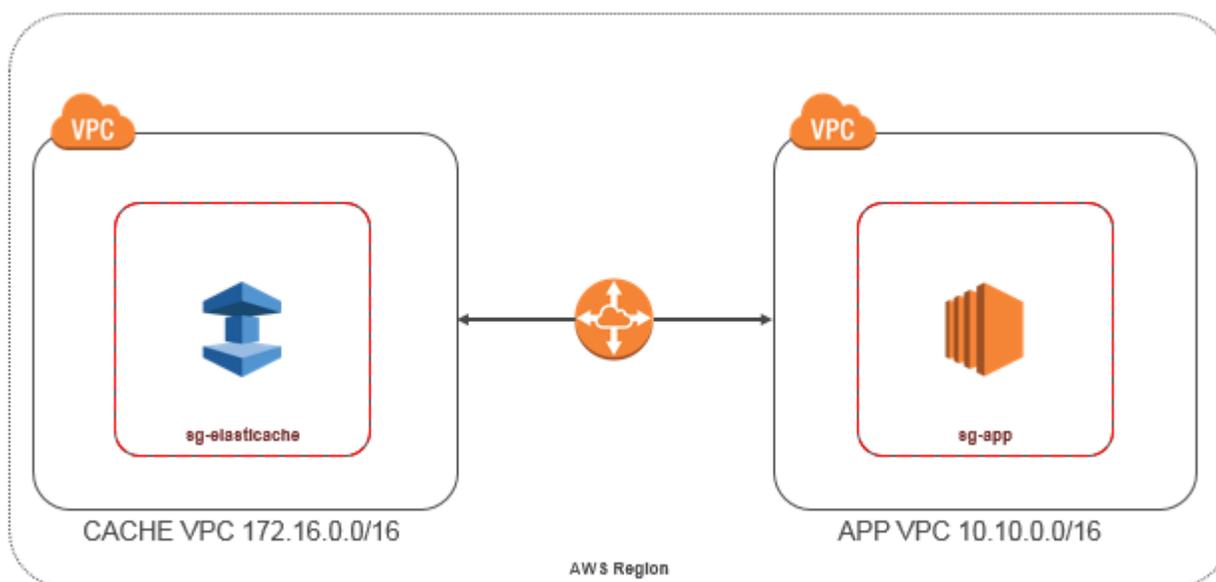
當您的快取與您用來存取該快取的 EC2 執行個體位於不同 VPC 中時，有幾種方式可存取快取。如果快取與 EC2 執行個體位於不同 VPC，但位於相同區域中，您可以使用 VPC 對等互連。如果快取與 EC2 執行個體位於不同區域，您可以在區域之間建立 VPC 連線。

#### 主題

- [在 ElastiCache 快取和 Amazon EC2 執行個體位於同一區域的不同 Amazon VPC 時存取快取](#)
- [在 ElastiCache 快取和 Amazon EC2 執行個體位於不同區域的不同 Amazon VPC 時存取快取](#)

在 ElastiCache 快取和 Amazon EC2 執行個體位於同一區域的不同 Amazon VPC 時存取快取

下圖說明使用 Amazon VPC 對等互連連線，透過相同區域的不同 Amazon VPC 中的 Amazon EC2 執行個體來存取快取。



## 由相同區域的不同 Amazon VPC 中的 Amazon EC2 執行個體存取的快取 - VPC 對等互連連線

VPC 對等連線是指兩個 VPC 之間的網路連線，透過此機制，您就可以使用私有 IP 地址在兩者之間路由流量。這兩個 VPC 中的執行個體能彼此通訊，有如位於相同網路中一樣。您可以在自己的 Amazon VPC 之間建立 VPC 對等連線，或使用單一區域內另一個 AWS 帳戶中的 Amazon VPC 建立 VPC 對等連線。若要進一步了解 Amazon VPC 對等互連，請參閱 [VPC 說明文件](#)。

### Note

對等 VPC 的 DNS 名稱解析可能會失敗，具體取決於套用至 VPC 的 ElastiCache 組態。若要解析名稱，兩端的 VPC 都必須啟用 DNS 主機名稱和 DNS 解析。如需詳細資訊，請參閱 [啟用 VPC 對等互連的 DNS 解析](#)。

透過對等互連存取不同 Amazon VPC 中的快取

1. 請確保兩個 VPC 沒有重疊的 IP 範圍，否則您將無法為其建立互連連線。
2. 為兩個 VPC 建立互連連線。如需詳細資訊，請參閱 [建立和接受 Amazon VPC 對等互連連線](#)。
3. 更新您的路由表。如需詳細資訊，請參閱 [更新 VPC 互連連線的路由表](#)

下列是上述圖表中用於示範的路由表。請注意，pcx-a894f1c1 是互連連線。

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 路由表

4. 修改 ElastiCache 快取的安全性群組，以允許來自對等 VPC 中「應用程式」安全性群組的輸入連線。如需詳細資訊，請參閱 [參考對等 VPC 安全群組](#)。

透過對等互連連線存取快取，將產生額外的資料傳輸費用。

### 使用 Transit Gateway

傳輸閘道可讓您在相同 AWS 區域附加 VPC 和 VPN 連線，並在它們之間路由流量。運輸閘道跨 AWS 帳戶運作，您可以使用 AWS Resource Access Manager 與其他帳戶共用傳輸閘道。在您與其他帳戶

共用交通閘道後，AWS 帳戶擁有者可以將其 VPC 附加到您的傳輸閘道。這些帳戶的使用者均可隨時刪除連接。

您可以在傳輸閘道上啟用多點傳送，然後建立傳輸閘道多點傳送網域，讓多點傳送流量可透過與網域相關聯的 VPC 連接，從多點傳送來源傳送至多點傳送群組成員。

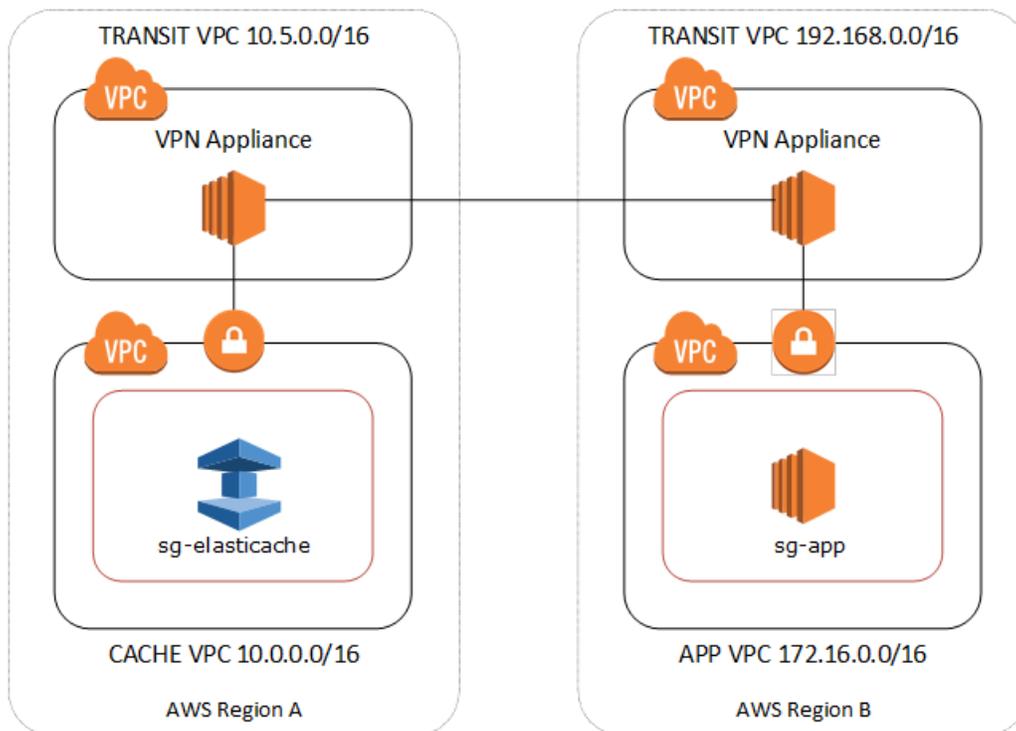
您也可以在不同 AWS 區域的傳輸閘道之間建立對等連線附件。這可讓您跨不同區域在傳輸閘道附件之間路由流量。

如需詳細資訊，請參閱[傳輸閘道](#)。

在 ElastiCache 快取和 Amazon EC2 執行個體位於不同區域的不同 Amazon VPC 時存取快取

## 使用傳輸 VPC

對於將多個在地理上分散的 VPC 和遠端網路進行連線，取代 VPC 對等互連的替代常用策略是建立做為全球網路傳輸中心的傳輸 VPC。傳輸 VPC 會簡化網路管理，並最大程度減少連線多個 VPC 和遠端網路所需的連線數。此設計可以節省時間和精力並降低費用，因為實際上不具有在託管傳輸中樞建立實體存在或部署實體網路設備的傳統支出。



## 位於不同區域不同 VPC 之間的連線

傳輸 Amazon VPC 建立後，部署在一個區域中「支點」VPC 中的應用程式可以連接到另一個區域內「支點」VPC 中的 ElastiCache 快取。

## 存取不同區域內不同 VPC 中的快取 AWS

1. 部署傳輸 VPC 解決方案。如需詳細資訊，請參閱 [AWS Transit Gateway](#)。
2. 更新應用程式和快取 VPC 中的 VPC 路由表，以透過 VGW (虛擬私有閘道) 和 VPN 設備來路由流量。在使用邊界閘道協定 (BGP) 的動態路由情況下，您的路由可能會自動傳播。
3. 修改 ElastiCache 快取的安全群組，以允許來自應用程式執行個體 IP 範圍的輸入連線。請注意，在此情況下，您將無法參考應用程式伺服器安全群組。

跨區域存取快取將造成網路延遲，並產生額外的跨區域資料傳輸費用。

### 從客戶資料中心執行的應用程式存取 ElastiCache 快取

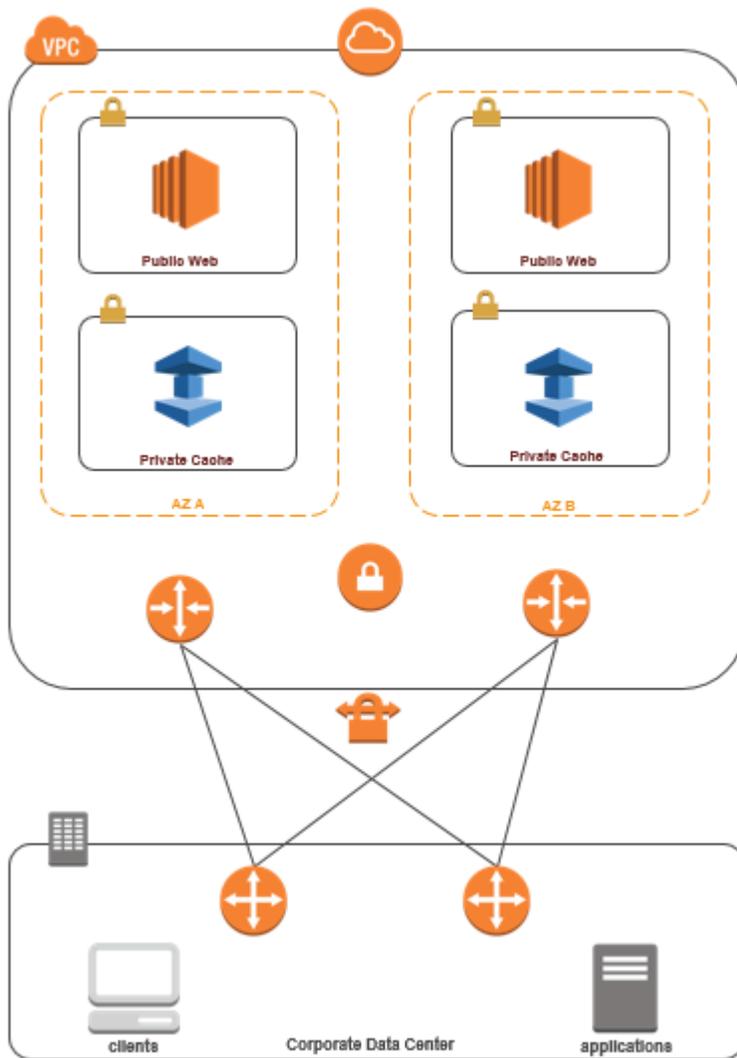
另一種可能的案例是混合式架構，客戶資料中心中的用戶端或應用程式可能需要存取 VPC 中的 ElastiCache 快取。如果客戶 VPC 與資料中心之間具有透過 VPN 或 Direct Connect 的連線，則此情況也受支援。

### 主題

- [使用 VPN 連線能力，從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)
- [使用直 Connect 從客戶資料中心執行的應用程式存取 ElastiCache 快取](#)

使用 VPN 連線能力，從客戶資料中心執行的應用程式存取 ElastiCache 快取

下圖說明使用 VPN 連線，從公司網路中執行的應用程式存取 ElastiCache 快取。



### 透過 VPN ElastiCache 從資料中心連線

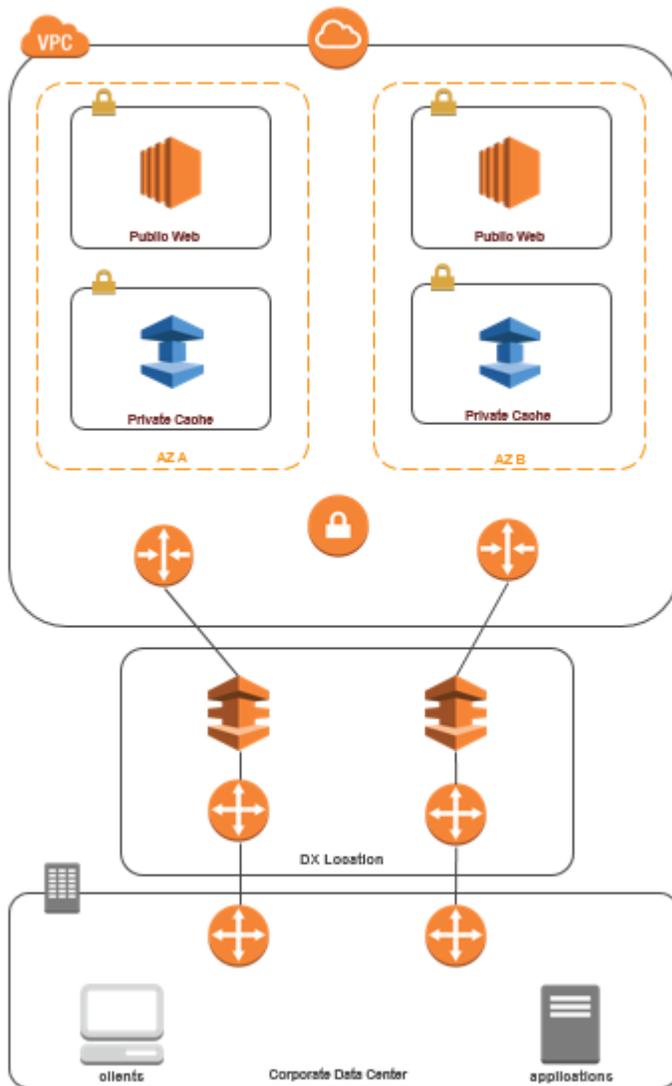
#### 透過 VPN 連線從內部部署應用程式存取 VPC 中的快取

1. 透過將硬體虛擬私有閘道新增至您的 VPC 來建立 VPN 連線。如需詳細資訊，請參閱[將硬體虛擬私有閘道新增至您的 VPC](#)。
2. 更新部署 ElastiCache 快取所在子網路的 VPC 路由表，以允許來自內部部署應用程式伺服器的流量。在使用 BGP 的動態路由情況下，您的路由可能會自動傳播。
3. 修改 ElastiCache 快取的安全性群組，以允許內部部署應用程式伺服器的輸入連線。

透過 VPN 連線存取快取將造成網路延遲，並產生額外的跨區域資料傳輸費用。

## 使用直 Connect 從客戶資料中心執行的應用程式存取 ElastiCache 快取

下圖說明使用 Direct Connect 從公司網路上執行的應用程式存取 ElastiCache 快取。



透過直接 Connect ElastiCache 從您的資料中心連接

使用「直 Connect」從網路中執行的應用程式存取 ElastiCache 快取

1. 建立 Direct Connect 連線。如需詳細資訊，請參閱[AWS 直 Connect 入門](#)。
2. 修改 ElastiCache 快取的安全性群組，以允許內部部署應用程式伺服器的輸入連線。

透過 DX 連線存取快取可能造成網路延遲，並產生額外的跨區域資料傳輸費用。

## 建立 Virtual Private Cloud (VPC)

在此範例中，您會建立 Amazon VPC，其中的每個可用區域具有一個私有子網路。

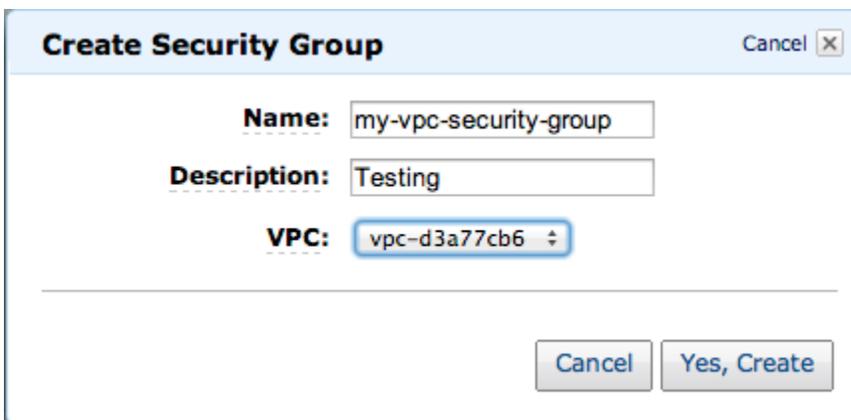
### 建立 Amazon VPC (主控台)

1. 登入 AWS 管理主控台，開啟位於 <https://console.aws.amazon.com/vpc/> 的 Amazon VPC 主控台。
2. 在 VPC 儀表中，選擇 Create VPC (建立 VPC)。
3. 在要建立的資源之下，選擇 VPC 等。
4. 在 Number of Availability Zones (AZs) (可用區域 (AZ) 數量) 中，選擇您要在其中啟動子網路的可用區域數量。
5. 在 Number of public subnets (公有子網路數量) 中，選擇您要新增至 VPC 的公用子網路數量。
6. 在 Number of private subnets (私有子網路數量) 中，選擇您要新增至 VPC 的私人子網路數量。

#### Tip

記下您的子網路識別符，分別為公有和私有。之後啟動叢集和將 Amazon EC2 執行個體新增到 Amazon VPC 時，您會需要此資訊。

7. 建立 Amazon VPC 安全群組。此群組將用於您的快取叢集和 Amazon EC2 執行個體。
  - a. 在 Amazon VPC 管理主控台的導覽窗格中，選擇 Security Groups (安全群組)。
  - b. 選擇建立安全群組。
  - c. 在對應的方塊中，為安全群組輸入名稱和描述。在 VPC 方塊中，選擇 Amazon VPC 的識別碼。



**Create Security Group** Cancel

**Name:** my-vpc-security-group

**Description:** Testing

**VPC:** vpc-d3a77cb6

Cancel Yes, Create

- d. 當您滿意設定後，請選擇 Yes, Create (是，建立)。

8. 為您的安全群組定義網路傳入規則。此規則將允許您使用 Secure Shell (SSH) 連接至 Amazon EC2 執行個體。
  - a. 在導覽清單中，選擇 Security Groups (安全群組)。
  - b. 在清單中找到您的安全群組，然後選擇它。
  - c. 在 Security Group (安全群組) 下，選擇 Inbound (入站) 標籤。在 Create a new rule (建立新規則) 方塊中，選擇 SSH，然後選擇 Add Rule (新增規則)。
  - d. 針對您的新傳入規則設定下列值，允許透過 HTTP 存取：
    - 類型：HTTP
    - 資源來源：0.0.0.0/0

選擇 Apply Rule Changes (套用規則變更)。

現在您已準備好在 Amazon VPC 中建立快取子網路群組和啟動快取叢集。

- [建立子網路群組](#)
- [建立 Memcached 叢集 \(主控台\)](#).

## 連線到 Amazon VPC 中執行的快取

此範例說明如何在 Amazon VPC 中啟動 Amazon EC2 執行個體。然後您可以登入此執行個體，並存取 Amazon VPC 中執行的 ElastiCache 快取。

### 連線到 Amazon VPC 中執行的快取 (主控台)

在此範例中，您需在 Amazon VPC 中建立 Amazon EC2 執行個體。您可以使用此 Amazon EC2 執行個體連線至 Amazon VPC 中執行的快取節點。

#### Note

如需有關使用 Amazon EC2 的資訊，請參閱 [Amazon EC2 說明文件](#) 中的 [Amazon EC2 入門指南](#)。

### 使用 Amazon EC2 主控台在 Amazon VPC 中建立 Amazon EC2 執行個體

1. 請登入 AWS Management Console，並在 <https://console.aws.amazon.com/ec2/> 開啟 Amazon EC2 主控台。
2. 在主控台中，選擇 Launch Instance (啟動執行個體)，然後依照下列步驟執行：
3. 在 Choose an Amazon Machine Image (AMI) (選擇 Amazon Machine Image (AMI)) 頁面上，選擇 64 位元 Amazon Linux AMI，然後選擇 Select (選取)。
4. 在 Choose an Instance Type (選擇執行個體類型) 頁面上選擇 3. Configure Instance (3. 設定執行個體)。
5. 在 Configure Instance Details (設定執行個體詳細資訊) 頁面上，選取下列選項：
  - a. 在 Network (網路) 清單中，選擇您的 Amazon VPC。
  - b. 在 Subnet (子網路) 清單中，選擇您的公有子網路。

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage pricing, assign an access management role to the instance, and more.

Number of instances ⓘ

Purchasing option ⓘ  Request Spot Instances

Network ⓘ

Subnet ⓘ    
250 IP Addresses available

Public IP ⓘ  Automatically assign a public IP address to your instances

當您滿意設定後，選擇 4. Add Storage (4. 新增儲存體)。

- 在 Add Storage (新增儲存體) 頁面上，選擇 5. Tag Instance (5. 標籤執行個體)。
- 在 Tag Instance (標記執行個體) 頁面上，輸入 Amazon EC2 執行個體的名稱，然後選擇 6. Configure Security Group (6. 設定安全群組)。
- 在 Configure Security Group (設定安全群組) 頁面上，選擇 Select an existing security group (選取現有的安全群組)。如需安全群組的詳細資訊，請參閱 [Linux 執行個體的 Amazon EC2 安全群組](#)。

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group:  Create a new security group  
 Select an existing security group

Security Group ID	Name	Description
<input type="checkbox"/> sg-1a3d2178	default	default VPC security group
<input checked="" type="checkbox"/> sg-f13d2193	my-vpc-security-group	Testing

選擇 Amazon VPC 安全群組的名稱，然後選擇 Review and Launch (檢閱並發佈)。

- 在 Review Instance and Launch (檢閱執行個體和啟動) 頁面上，選擇 Launch (啟動)。
- 在 Select an existing key pair or create a new key pair (選取現有的金鑰配對或建立新的金鑰配對) 視窗中，指定您要用於此執行個體的金鑰對。

**Note**

如需有關管理金鑰對的資訊，請參閱 [Amazon EC2 入門指南](#)。

11. 當您準備好啟動 Amazon EC2 執行個體時，請選擇 Launch (啟動)。

您現在可以將彈性 IP 地址指派給剛才建立的 Amazon EC2 執行個體。您必須使用此 IP 地址才能連線到 Amazon EC2 執行個體。

### 指派彈性 IP 地址 (主控台)

1. 在 <https://console.aws.amazon.com/vpc/> 開啟 Amazon VPC 主控台。
2. 在導覽清單中，選擇 Elastic IPs (彈性 IP)。
3. 選擇 Allocate Elastic IP address (配置彈性 IP 地址)。
4. 在 Allocate Elastic IP address (配置彈性 IP 地址) 對話方塊中，接受預設的 Network Border Group (網路邊界群組)，然後選擇 Allocate (配置)。
5. 從清單中選擇您剛才配置的彈性 IP 地址，然後選擇 Associate Address (關聯地址)。
6. 在 Associate Address (建立地址關聯) 對話方塊的 Instance (執行個體) 方塊中，選擇您啟動之 Amazon EC2 執行個體的 ID。

在 Private IP address (私有 IP 地址) 方塊中，選取該方塊以取得私有 IP 地址，然後選擇 Associate (關聯)。

您現在可以使用 SSH 連線到您所建立使用彈性 IP 地址的 Amazon EC2 執行個體。

### 連線到您的 Amazon EC2 執行個體

- 開啟命令視窗。在命令提示中，發出以下命令，以您的金鑰對檔案名稱取代 mykeypair.pem，以您的彈性 IP 地址取代 54.207.55.251。

```
ssh -i mykeypair.pem ec2-user@54.207.55.251
```

**Important**

此時請勿登出您的 Amazon EC2 執行個體。

您現在已準備好與 ElastiCache 叢集進行互動。在您可以這樣做之前 (如果您還沒有這樣做)，您必須安裝 telnet 公用程式。

### 安裝 Telnet 並與您的快取叢集互動 (AWS CLI)

1. 開啟命令視窗。在命令提示中，發出下列命令。在確認提示中，輸入 y。

```
sudo yum install telnet
Loaded plugins: priorities, security, update-motd, upgrade-helper
Setting up Install Process
Resolving Dependencies
--> Running transaction check

...(output omitted)...

Total download size: 63 k
Installed size: 109 k
Is this ok [y/N]: y
Downloading Packages:
telnet-0.17-47.7.amzn1.x86_64.rpm                | 63 kB      00:00

...(output omitted)...

Complete!
```

2. 前往 ElastiCache 主控台 (<https://console.aws.amazon.com/elasticache/>)，並取得快取叢集中其中一個節點的端點。如需詳細資訊，請參閱適用於 Memcached 的[尋找連線端點](#)。
3. 使用 Telnet 透過連接埠 11211 連線到快取節點端點。以您的快取節點主機名稱取代以下顯示的主機名稱。

```
telnet my-cache-cluster.7wufxa.0001.use1.cache.amazonaws.com 11211
```

您現在已連接到快取引擎，並可以發出命令。在此範例中，您新增一個資料項目到快取，然後立即取得它。最後，您將中斷與快取節點的連線。

若要儲存金鑰和值，請輸入下列兩行：

```
add mykey 0 3600 28
This is the value for mykey
```

快取引擎會以下列方式回應：

```
OK
```

若要擷取 mykey 的值，請輸入下列命令：

```
get mykey
```

快取引擎會以下列方式回應：

```
VALUE mykey 0 28  
This is the value for my key  
END
```

若要中斷與快取引擎的連線，請輸入下列命令：

```
quit
```

#### Important

若要避免對 AWS 帳戶衍生額外費用，在嘗試這些範例後，請務必刪除您不再需要的任何 AWS 資源。

## Amazon ElastiCache API 和界面 VPC 端點 (AWS PrivateLink)

您可以建立界面 VPC 端點，以在您的 VPC 與 Amazon ElastiCache API 端點之間建立私有連線。界面端點採用 [AWS PrivateLink](#)。AWS PrivateLink 可讓您可以私有方式存取 Amazon ElastiCache API 操作，無需網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線。

VPC 中的執行個體不需要公有 IP 地址，即能與 Amazon ElastiCache API 端點通訊。您的執行個體也不需要公有 IP 地址，就能使用任何可用的 ElastiCache API 操作。您的 VPC 與 Amazon ElastiCache 之間的流量，都會在 Amazon 網路的範圍內。每個界面端點都是由您子網路中的一或多個彈性網路界面表示。如需彈性網路界面的詳細資訊，請參閱 Amazon EC2 使用者指南中的 [彈性網路界面](#)。

- 如需 VPC 端點的詳細資訊，請參閱 Amazon VPC 使用者指南中的 [界面 VPC 端點 \(AWS PrivateLink\)](#)。

- 如需 ElastiCache API 操作的詳細資訊，請參閱 [ElastiCache API 操作](#)。

在建立界面 VPC 端點之後，如果您針對端點啟用[私有 DNS](#) 主機名稱，則預設 ElastiCache 端點 (<https://elasticache.Region.amazonaws.com>) 會解析為您的 VPC 端點。如果您尚未啟用私有 DNS 主機名稱，Amazon VPC 會透過以下格式提供一個 DNS 端點名稱，供您使用：

```
VPC_Endpoint_ID.elasticache.Region.vpce.amazonaws.com
```

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[介面 VPC 端點 \(AWS PrivateLink\)](#)。ElastiCache 支援在您的 VPC 內呼叫其所有 [API 動作](#)。

#### Note

只能為 VPC 中的一個 VPC 端點啟用私人 DNS 主機名稱。如果您想要建立其他 VPC 端點，則應停用該端點的私人 DNS 主機名稱。

## VPC 端點的考量事項

在設定 Amazon ElastiCache API 端點的界面 VPC 端點前，請務必參閱《Amazon VPC 使用者指南》中的[介面端點屬性和限制](#)。所有與管理 Amazon ElastiCache 資源相關的 ElastiCache API 操作，都可從使用 AWS PrivateLink 的 VPC 使用。

ElastiCache API 端點支援 VPC 端點政策。預設允許透過端點完整存取 ElastiCache API 操作。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[使用 VPC 端點控制對服務的存取](#)。

## 為 ElastiCache API 建立界面 VPC 端點

您可使用 Amazon VPC 主控台或 AWS CLI，為 Amazon ElastiCache API 建立 VPC 端點。如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[建立介面端點](#)。

在您建立界面 VPC 端點後，您可以為該端點啟用私有 DNS 主機名稱。啟用後，預設 Amazon ElastiCache 端點 (<https://elasticache.Region.amazonaws.com>) 會解析為您的 VPC 端點。對於中國 (北京) 和中國 (寧夏) AWS 區域，您可以針對北京使用 [elasticache.cn-north-1.amazonaws.com.cn](https://elasticache.cn-north-1.amazonaws.com.cn) 和針對寧夏使用 [elasticache.cn-northwest-1.amazonaws.com.cn](https://elasticache.cn-northwest-1.amazonaws.com.cn)，對 VPC 端點進行 API 請求。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[透過介面端點存取服務](#)。

## 為 Amazon ElastiCache API 建立 VPC 端點政策

您可以將端點政策連接至控制 ElastiCache API 存取權限的 VPC 端點。此政策會指定以下項目：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[使用 VPC 端點控制對服務的存取](#)。

### Example ElastiCache API 動作的 VPC 端點政策

以下是 ElastiCache API 端點政策的範例。連接至端點後，此政策會針對所有資源上的所有主體，授予列出的 ElastiCache API 動作的存取權限。

```
{
  "Statement": [{
    "Principal": "*",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster",
      "elasticache:ModifyCacheCluster"
    ],
    "Resource": "*"
  }]
}
```

### Example 拒絕所有來自指定 AWS 帳戶之存取的 VPC 端點政策

下列 VPC 端點政策拒絕 AWS 帳戶 **123456789012** 對使用該端點之資源的任何存取。此政策允許來自其他帳戶的所有動作。

```
{
  "Statement": [{
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*",
    "Principal": "*"
  }],
  {
    "Action": "*",
```

```
"Effect": "Deny",
"Resource": "*",
"Principal": {
  "AWS": [
    "123456789012"
  ]
}
]
```

## 子網路和子網路群組

子網路群組是子網路的集合 (一般是私有)，您可以為 Amazon Virtual Private Cloud (VPC) 環境中執行的自行設計叢集指定這些子網路。

如果您在 Amazon VPC 中建立自行設計的叢集，則必須使用子網路群組。ElastiCache 會使用此子網路群組選擇子網路及該子網路中的 IP 地址，以與您的節點建立關聯。

ElastiCache 提供預設 IPv4 子網路群組，或者您可以選擇建立新的子網路群組。針對 IPv6，您必須建立具有 IPv6 CIDR 區塊的子網路群組。若您選擇 dual stack (雙堆疊)，則須選取 IPv6 或 IPv4 做為探索 IP 類型。

ElastiCache Serverless 不會使用子網路群組資源，而是在建立期間直接取得子網路清單。

本節說明如何建立和利用子網路和子網路群組來管理 ElastiCache 資源的存取權。

如需 Amazon VPC 環境中子網路群組使用方式的詳細資訊，請參閱「[存取您的叢集](#)」。

### 主題

- [建立子網路群組](#)
- [將子網路群組指派至快取](#)
- [修改子網路群組](#)
- [刪除子網路群組](#)

## 建立子網路群組

快取子網路群組是子網路的集合，您可能想要在 VPC 中為快取指定該集合。在 VPC 中啟動快取時，您必須選取快取子網路群組。然後 ElastiCache 會使用該快取子網路群組，對快取中的每個快取節點指派該子網路內的 IP 地址。

建立新子網路群組時，請記下可用 IP 地址的數量。如果子網路擁有的可用 IP 地址很少，您還可以新增至叢集的節點數量可能會受到限制。若要解決此問題，您可以對子網路群組指定一或多個子網路，使得您在叢集的可用區域中有足夠數量的 IP 地址。在那之後，您便可以將更多節點新增至您的叢集。

若您選擇 IPv4 為網路類型，則可用預設子網路群組，或您也可以選擇建立新的子網路群組。ElastiCache 會使用此子網路群組選擇子網路及該子網路中的 IP 地址，以與您的節點建立關聯。若您選擇雙堆疊或 IPv6，系統會引導您建立雙堆疊或 IPv6 子網路。如需網路類型的詳細資訊，請參閱[網路類型](#)。如需詳細資訊，請參閱[在您的 VPC 中建立子網路](#)。

下列程序顯示如何使用主控台、AWS CLI 和 ElastiCache API 來建立名為 mysubnetgroup 的子網路群組。

### 建立子網路群組 (主控台)

下列程序顯示如何建立子網路群組 (主控台)。

### 建立子網路群組 (主控台)

1. 登入 AWS 管理主控台，並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽清單中，選擇 Subnet Groups (子網路群組)。
3. 選擇 Create subnet group (建立子網路群組)。
4. 在 Create Subnet Group (建立子網路群組) 精靈中，執行下列動作。依您需要完成所有設定後，選擇建立。
  - a. 在 Name (名稱) 方塊中，輸入子網路群組的名稱。
  - b. 在 Description (描述) 方塊中，輸入子網路群組的描述。
  - c. 在 VPC ID 方塊中，選擇您的 Amazon VPC。
  - d. 所有子網路都是預設選擇的狀態。在選取子網路面板中，按一下管理，並選取私有子網路的可用區域或[本地區域](#)和 ID，然後選擇選擇。
5. 在出現的確認訊息中，選擇 Close (關閉)。

您新的子網路群組會顯示在 ElastiCache 主控台的 Subnet Groups (子網路群組) 清單中。您可以在視窗底部選擇要查看詳細資訊的子網路群組，例如與此群組相關聯的所有子網路。

### 建立子網路群組 (AWS CLI)

在命令提示字元中，使用命令 `create-cache-subnet-group` 來建立子網路群組。

若為 Linux、macOS 或 Unix：

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

針對 Windows：

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

此命令應該產生類似下列的輸出：

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

如需詳細資訊，請參閱 AWS CLI 主題 [create-cache-subnet-group](#)。

## 將子網路群組指派至快取

建立子網路群組之後，您可以在 Amazon VPC 中啟動快取。如需更多資訊，請參閱下列內容。

- Memcached 叢集 - 若要啟動 Memcached 叢集，請參閱「[建立 Memcached 叢集 \(主控台\)](#)」。在步驟 7.a (進階 Memcached 設定) 中，選擇 VPC 子網路群組。

## 修改子網路群組

您可以修改子網路群組的描述，或修改與子網路群組關聯之子網路 ID 的清單。如果快取目前正在使用子網路，則無法從該子網路群組刪除該子網路 ID。

下列程序顯示如何修改子網路群組。

### 修改子網路群組 (主控台)

#### 修改子網路群組

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。
3. 在子網路群組清單中，選擇您要修改的子網路群組的選項按鈕，然後選擇修改。
4. 在選取之子網路面板中，選擇管理。
5. 對選取之子網路進行任何變更，然後按一下選擇。
6. 按一下儲存變更，儲存您的變更。

### 修改子網路群組 (AWS CLI)

在命令提示字元中，使用命令 `modify-cache-subnet-group` 來修改子網路群組。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup \  
  --cache-subnet-group-description "New description" \  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

針對 Windows：

```
aws elasticache modify-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup ^  
  --cache-subnet-group-description "New description" ^  
  --subnet-ids "subnet-42df9c3a" "subnet-48fc21a9"
```

此命令應該產生類似下列的輸出：

```
{
  "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"
  }
}
```

如需詳細資訊，請參閱 AWS CLI 主題 [modify-cache-subnet-group](#)。

## 刪除子網路群組

如果您決定不再需要使用您的子網路群組，您可以將它刪除。如果有某個快取正在使用子網路群組，則無法將它刪除。

下列程序顯示如何刪除子網路群組。

### 刪除子網路群組 (主控台)

#### 刪除子網路群組

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在導覽窗格中選擇 Subnet groups (子網路群組)。
3. 在子網路群組的清單中，選擇您要刪除的子網路群組，然後選擇 Delete (刪除)。
4. 當系統要求您確認此操作時，請在文字輸入欄位中輸入子網路群組的名稱，然後選擇刪除。

### 刪除子網路群組 (AWS CLI)

使用 AWS CLI，搭配下列參數呼叫命令 `delete-cache-subnet-group`：

- `--cache-subnet-group-name` *mysubnetgroup*

若為 Linux、macOS 或 Unix：

```
aws elasticache delete-cache-subnet-group \  
  --cache-subnet-group-name mysubnetgroup
```

針對 Windows：

```
aws elasticache delete-cache-subnet-group ^  
  --cache-subnet-group-name mysubnetgroup
```

此命令不會產生輸出。

如需詳細資訊，請參閱 AWS CLI 主題 [delete-cache-subnet-group](#)。

## Amazon Identity and Access Management ElastiCache

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。AWS 服務 IAM 管理員控制哪些人可以驗證 (登入) 和授權 (具有權限) 以使用 ElastiCache 資源。您可以使用 IAM AWS 服務，無需額外付費。

## 主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [Amazon 如何與 IAM 合 ElastiCache 作](#)
- [Amazon ElastiCache 的身分型政策範例](#)
- [疑難排解 Amazon ElastiCache 身分和存取](#)
- [存取控制](#)
- [管理 ElastiCache 資源存取許可的概觀](#)

## 物件

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，具體取決於您在進行的工作 ElastiCache。

**服務使用者** — 如果您使用 ElastiCache 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 ElastiCache 功能來完成工作時，您可能需要其他權限。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果無法存取中的圖徵 ElastiCache，請參閱[疑難排解 Amazon ElastiCache 身分和存取](#)。

**服務管理員** — 如果您負責公司的 ElastiCache 資源，您可能擁有完整的存取權 ElastiCache。決定您的服務使用者應該存取哪些 ElastiCache 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步瞭解貴公司如何搭配使用 IAM ElastiCache，請參閱[Amazon 如何與 IAM 合 ElastiCache 作](#)。

**IAM 管理員** — 如果您是 IAM 管理員，您可能想要瞭解如何撰寫政策來管理存取權限的詳細資訊 ElastiCache。若要檢視可在 IAM 中使用的 ElastiCache 基於身分的政策範例，請參閱。[Amazon ElastiCache 的身分型政策範例](#)

## 使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱 [AWS 登入 使用者指南中的如何登入您 AWS 帳戶](#) 的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的 [簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的 [多重要素驗證](#) 和 IAM 使用者指南中的 [在 AWS 中使用多重要素驗證 \(MFA\)](#)。

## AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的 [需要根使用者憑證的任務](#)。

## 聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時認證 AWS 服務 來存取。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identity Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務 的任何使用者。AWS Directory Service 同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的 [什麼是 IAM Identity Center ?](#)。

## IAM 使用者和群組

[IAM 使用者](#)是您內部的身份，具 AWS 帳戶有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 [IAM 使用者指南](#)中的為需要長期憑證的使用案例定期輪換存取金鑰。

[IAM 群組](#)是一種指定 IAM 使用者集合的身份。您無法以群組身份簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱IAM 使用者指南中的[建立 IAM 使用者 \(而非角色\) 的時機](#)。

## IAM 角色

[IAM 角色](#)是您 AWS 帳戶 內部具有特定許可的身份。它類似 IAM 使用者，但不與特定的人員相關聯。您可以[切換角色，在中暫時擔任 IAM 角色](#)。AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱 IAM 使用者指南中的[使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身份使用者存取 — 如需向聯合身份指派許可，請建立角色，並為角色定義許可。當聯合身份進行身份驗證時，該身份會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 [IAM 使用者指南](#)中的為第三方身份提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身份驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的[許可集](#)。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源 (而不是使用角色作為代理)。若要了解跨帳戶存取角色和以資源為基礎的政策之間的差異，請參閱 IAM 使用者指南中的 [IAM 中的跨帳戶資源存取](#)。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。

- 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。
- 服務角色 – 服務角色是服務擔任的 [IAM 角色](#)，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務服務](#)。
- 服務連結角色 — 服務連結角色是連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需詳細資訊，請參閱 IAM 使用者指南中的 [利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

如需了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的 [建立 IAM 角色 \(而非使用者\) 的時機](#)。

## 使用政策管理存取權

您可以透過 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS，當與身分識別或資源相關聯時，會定義其權限。AWS 當主參與者 (使用者、root 使用者或角色工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會 AWS 以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 [JSON 政策概觀](#)。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console AWS CLI、或 AWS API 取得角色資訊。

## 身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的[建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶。受管政策包括 AWS 受管政策和客戶管理的策略。如需了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的[在受管政策和內嵌政策間選擇](#)。

## 資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

## 存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。AWS WAF 如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的[存取控制清單 \(ACL\) 概觀](#)。

## 其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政

策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 [IAM 實體許可界限](#)。

- 服務控制策略 ( SCP ) — SCP 是 JSON 策略，用於指定中組織或組織單位 ( OU ) 的最大權限。AWS Organizations 是一種用於分組和集中管理您企業擁有的多個 AWS 帳戶的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需 Organizations 和 SCP 的詳細資訊，請參閱 AWS Organizations 使用者指南中的 [SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的 [工作階段政策](#)。

## 多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的 [政策評估邏輯](#)。

## Amazon 如何與 IAM 合 ElastiCache 作

在您使用 IAM 管理存取權限之前 ElastiCache，請先了解哪些 IAM 功能可搭配使用 ElastiCache。

您可以與 Amazon 搭配使用的 IAM 功能 ElastiCache

IAM 功能	ElastiCache 支持
<a href="#">身分型政策</a>	是
<a href="#">資源型政策</a>	否
<a href="#">政策動作</a>	是
<a href="#">政策資源</a>	是
<a href="#">政策條件索引鍵</a>	是
<a href="#">ACL</a>	是
<a href="#">ABAC (政策中的標籤)</a>	是

IAM 功能	ElastiCache 支持
<a href="#">臨時憑證</a>	是
<a href="#">主體許可</a>	是
<a href="#">服務角色</a>	是
<a href="#">服務連結角色</a>	是

若要深入瞭解如何以 ElastiCache 及其他 AWS 服務如何使用大多數 IAM 功能，請參閱 IAM 使用者指南中的搭配 IAM 使用的[AWS 服務](#)。

## 以身分識別為基礎的原則 ElastiCache

支援身分型政策	是
---------	---

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的[IAM JSON 政策元素參考](#)。

### ElastiCache 的身分型政策範例

若要檢視以 ElastiCache 身為基礎的原則範例，請參閱。[Amazon ElastiCache 的身分型政策範例](#)

## ElastiCache 內的資源型政策

支援以資源基礎的政策	否
------------	---

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源

的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

如需啟用跨帳戶存取權，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱 IAM 使用者指南中的[IAM 中的跨帳戶資源存取](#)。

## 的政策動作 ElastiCache

支援政策動作 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 ElastiCache 動作清單，請參閱服務授權參考 ElastiCache 中[由 Amazon 定義的動作](#)。

中的策略動作在動作之前 ElastiCache 使用下列前置詞：

```
elasticache
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [  
  "elasticache:action1",  
  "elasticache:action2"  
]
```

您也可以使用萬用字元 (\*) 來指定多個動作。例如，若要指定開頭是 Describe 文字的所有動作，請包含以下動作：

```
"Action": "elasticache:Describe*"
```

若要檢視以 ElastiCache 身為基礎的原則範例，請參閱 [Amazon ElastiCache 的身分型政策範例](#) 的政策資源 ElastiCache

支援政策資源

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (\*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

要查看 ElastiCache 資源類型及其 ARN 的列表，請參閱服務授權參考 ElastiCache 中 [由 Amazon 定義的資源](#)。若要了解可以使用哪些動作指定每個資源的 ARN，請參閱 [Amazon ElastiCache 定義的動作](#)。

若要檢視以 ElastiCache 身為基礎的原則範例，請參閱 [Amazon ElastiCache 的身分型政策範例](#)

ElastiCache 的政策條件索引鍵

支援服務特定政策條件金鑰

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用[條件運算子](#)的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯 OR 運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的[IAM 政策元素：變數和標籤](#)。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的[AWS 全域條件內容金鑰](#)。

若要查看 ElastiCache 條件金鑰清單，請參閱服務授權參考 ElastiCache 中的[Amazon 條件金鑰](#)。若要了解您可以使用條件金鑰的動作和資源，請參閱[Amazon 定義的動作 ElastiCache](#)。

若要檢視以 ElastiCache 身為基礎的原則範例，請參閱。[Amazon ElastiCache 的身分型政策範例](#)

## 存取控制清單 (ACL) ElastiCache

支援 ACL	是
--------	---

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

## 以屬性為基礎的存取控制 (ABAC) 搭配 ElastiCache

支援 ABAC (政策中的標籤)	是
------------------	---

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的[條件元素](#)中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱 IAM 使用者指南中的[什麼是 ABAC?](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱 IAM 使用者指南中的[使用屬性型存取控制 \(ABAC\)](#)。

## 使用臨時登入資料 ElastiCache

支援臨時憑證 是

當您使用臨時憑據登錄時，某些 AWS 服務 不起作用。如需其他資訊，包括哪些 AWS 服務 與臨時登入資料[搭配AWS 服務 使用](#)，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱 IAM 使用者指南中的[切換至角色 \(主控台\)](#)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱[IAM 中的暫時性安全憑證](#)。

## ElastiCache 的跨服務主體權限

支援轉寄存取工作階段 (FAS) 是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。

## 的服務角色 ElastiCache

支援服務角色 是

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務服務](#)。

### Warning

變更服務角色的權限可能會中斷 ElastiCache 功能。只有在 ElastiCache 提供指引時才編輯服務角色。

## 服務連結角色 ElastiCache

支援服務連結角色	是
----------	---

服務連結角色是一種連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服务連結角色的詳細資訊，請參閱 [可搭配 IAM 運作的 AWS 服務](#)。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

## Amazon ElastiCache 的身分型政策範例

根據預設，使用者和角色不具備建立或修改 ElastiCache 資源的許可。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 執行任務。若要授與使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的 [建立 IAM 政策](#)。

如需 ElastiCache 所定義之動作和資源類型的詳細資訊，包括每種資源類型的 ARN 格式，請參閱《服務授權參考》中的 [Amazon ElastiCache 適用的動作、資源和條件索引鍵](#)。

### 主題

- [政策最佳實務](#)
- [使用 ElastiCache 主控台](#)
- [允許使用者檢視他們自己的許可](#)

## 政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 ElastiCache 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並朝向最低權限許可的目標邁進：如需開始授予許可給使用者和工作負載，請使用 AWS 受管政策，這些政策會授予許可給許多常用案例。它們可在您的 AWS 帳戶中使用。我們建議您定義特定於使用案例的 AWS 客戶管理政策，以便進一步減少許可。如需更多資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授予對服務動作的存取權，前提是透過特定 AWS 服務（例如 AWS CloudFormation）使用條件。如需更多資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。
- 需要多重要素驗證 (MFA)：如果存在需要 AWS 帳戶中 IAM 使用者或根使用者的情況，請開啟 MFA 提供額外的安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

有關 IAM 中最佳實務的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 最佳安全實務](#)。

## 使用 ElastiCache 主控台

若要存取 Amazon ElastiCache 主控台，您必須擁有最基本的一組許可。這些許可必須允許您列出和檢視您 AWS 帳戶中 ElastiCache 資源的詳細資訊。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體（使用者或角色）而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許其最基本主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

為確保使用者和角色仍可使用 ElastiCache 主控台，還請將 ElastiCache ConsoleAccess 或 ReadOnly AWS 受管政策連接至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [新增許可到使用者](#)。

## 允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台上，或是使用 AWS CLI 或 AWS API 透過編寫程式的方式完成此動作的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

## 疑難排解 Amazon ElastiCache 身分和存取

使用下列資訊可協助您診斷和修正使用和 IAM 時可能會遇到的 ElastiCache 常見問題。

## 主題

- [我沒有執行操作的授權 ElastiCache](#)
- [我沒有授權執行 iam : PassRole](#)
- [我想允許 AWS 帳戶以外的人員存取我的 ElastiCache 資源](#)

### 我沒有執行操作的授權 ElastiCache

如果 AWS Management Console 告訴您您沒有執行動作的授權，則您必須聯絡您的管理員以尋求協助。您的管理員是提供您使用者名稱和密碼的人員。

下列範例錯誤會在 mateojackson 使用者嘗試使用主控台檢視一個虛構 *my-example-widget* 資源的詳細資訊，但卻無虛構 `elasticache:GetWidget` 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
elasticache:GetWidget on resource: my-example-widget
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-example-widget* 動作存取 `elasticache:GetWidget` 資源。

### 我沒有授權執行 iam : PassRole

如果您收到未獲授權執行 `iam:PassRole` 動作的錯誤訊息，則必須更新您的原則以允許您將角色傳遞給 ElastiCache。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM 使用者 marymajor 嘗試使用主控台執行中的動作時，會發生下列範例錯誤 ElastiCache。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

## 我想允許 AWS 帳戶以外的人員存取我的 ElastiCache 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解是否 ElastiCache 支援這些功能，請參閱[Amazon 如何與 IAM 合 ElastiCache 作](#)。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱 [IAM 使用者指南中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限](#)。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的[提供第三方 AWS 帳戶 擁有的存取權](#)。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的[將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 若要了解跨帳戶存取使用角色和以資源為基礎的政策之間的差異，請參閱 IAM 使用者指南中的 [IAM 中的跨帳戶資源存取](#)。

## 存取控制

您可以擁有有效的認證來驗證您的請求，但除非您擁有權限，否則您無法建立或存取 ElastiCache 資源。例如，您必須具有建立 ElastiCache 叢集的權限。

下列各節說明如何管理的權限 ElastiCache。我們建議您先閱讀概觀。

- [管理 ElastiCache 資源存取許可的概觀](#)
- [針對 Amazon ElastiCache 使用身分型政策 \(IAM 政策\)](#)

## 管理 ElastiCache 資源存取許可的概觀

每項 AWS 資源均由某個 AWS 帳戶所持有，而建立或存取資源的許可則由許可政策管理。帳戶管理員可以將許可政策連接到 IAM 身分 (即使用者、群組和角色)。此外，Amazon ElastiCache 還支援將許可政策連接到資源。

### Note

帳戶管理員 (或管理員使用者) 是具有管理員權限的使用者。如需詳細資訊，請參 [《IAM 使用者指南》](#) 中的 IAM 最佳實務。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- AWS IAM Identity Center 中的使用者和群組：

建立權限合集。請遵循 AWS IAM Identity Center 使用者指南的 [建立權限合集](#) 中的指示。

- 透過身分提供者在 IAM 中管理的使用者：

建立聯合身分的角色。請遵循 IAM 使用者指南的 [為第三方身分提供者 \(聯合\) 建立角色](#) 中的指示。

- IAM 使用者：

- 建立您的使用者可擔任的角色。請遵循 IAM 使用者指南的 [為 IAM 使用者建立角色](#) 中的指示。
- (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的 [新增權限至使用者 \(主控台\)](#) 中的指示。

### 主題

- [Amazon ElastiCache 資源和作業](#)
- [了解資源所有權](#)
- [管理資源存取](#)
- [Amazon ElastiCache 的 AWS 受管政策](#)
- [針對 Amazon ElastiCache 使用身分型政策 \(IAM 政策\)](#)
- [資源層級許可](#)
- [使用條件索引鍵](#)
- [為 Amazon ElastiCache 使用服務連結角色](#)

- [ElastiCache API 權限：動作、資源和條件參考](#)

## Amazon ElastiCache 資源和作業

若要查看 ElastiCache 資源類型清單及其 ARN，請參閱《服務授權參考》中的 [Amazon ElastiCache 定義的資源](#)。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 [Amazon ElastiCache 定義的動作](#)。

### 了解資源所有權

資源擁有者是建立資源的 AWS 帳戶。換言之，資源擁有者就是驗證建立資源請求之委託人實體的 AWS 帳戶。委託人實體可以是根帳戶、IAM 使用者或 IAM 角色)。下列範例說明其如何運作：

- 假設您使用 AWS 帳戶的根帳戶憑證來建立快取叢集。在此例中，您的 AWS 帳戶即為資源的擁有者。在 ElastiCache 中，資源就是快取叢集。
- 假設您在 AWS 帳戶中建立 IAM 使用者並將建立快取叢集的許可授予該使用者。在此例中，該使用者可以建立快取叢集。但是您的 AWS 帳戶 (也是該使用者所屬的帳戶) 擁有該快取叢集資源。
- 假設您在 AWS 帳戶中建立具有建立快取叢集許可的 IAM 角色。在此例中，任何可以擔任該角色的人都能建立快取叢集。您的 AWS 帳戶 (也是該角色所屬的帳戶) 擁有快取叢集資源。

### 管理資源存取

許可政策描述誰可以存取哪些資源。下一節說明可用來建立許可政策的選項。

#### Note

本節著重討論如何在 Amazon ElastiCache 的環境中使用 IAM。它不提供 IAM 服務的詳細資訊。如需完整的 IAM 文件，請參閱 IAM 使用者指南中的 [什麼是 IAM](#)。如需有關 IAM 政策語法和說明的資訊，請參閱 IAM 使用者指南中的 [AWS IAM 政策參考](#)。

連接到 IAM 身分的政策稱為身分類型政策 (IAM 政策)。連接到資源的政策稱為資源型政策。

#### 主題

- [身分類型政策 \(IAM 政策\)](#)
- [指定政策元素：動作、效果、資源和主體](#)

- [在政策中指定條件](#)

## 身分類型政策 (IAM 政策)

您可以將政策連接到 IAM 身分。例如，您可以執行下列動作：

- 將許可政策連接至帳戶中的使用者或群組 - 帳戶管理員能夠透過與特定使用者相關聯的許可政策來授予許可。在此案例中，許可用於授予該使用者建立 ElastiCache 資源，例如快取叢集、參數群組或安全群組。
- 將許可政策連接至角色 (授予跨帳戶許可)：您可以將身分識別型許可政策連接至 IAM 角色，藉此授予跨帳戶許可。例如，帳戶 A 管理員可以建立角色，將跨帳戶許可授予另一個 AWS 帳戶 (例如帳戶 B) 或某個 AWS 服務，如下所示：
  1. 帳戶 A 管理員建立 IAM 角色，並將許可政策連接到可授與帳戶 A 中資源許可的角色。
  2. 帳戶 A 管理員將信任政策連接至該角色，識別帳戶 B 做為可擔任該角的委託人。
  3. 帳戶 B 管理員即可將擔任該角色的許可委派給帳戶 B 中的任何使用者。這麼做可讓帳戶 B 的使用者建立或存取帳戶 A 的資源。有時候，您可能會想要授與 AWS 服務許可以擔任該角色。為了支援此方法，信任政策中的委託人也可以是 AWS 服務委託人。

如需使用 IAM 來委派許可的詳細資訊，請參閱《IAM 使用者指南》中的[存取管理](#)。

以下為允許使用者對您的 AWS 帳戶執行 DescribeCacheClusters 動作的範例政策。ElastiCache 還支援使用 API 動作的資源 ARN 來識別特定資源。(此方法也稱為資源層級許可)。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DescribeCacheClusters",
    "Effect": "Allow",
    "Action": [
      "elasticache:DescribeCacheClusters"],
    "Resource": resource-arn
  ]
}
```

如需搭配 ElastiCache 使用身分型政策的詳細資訊，請參閱「[針對 Amazon ElastiCache 使用身分型政策 \(IAM 政策\)](#)」。如需使用者、群組、角色和許可的詳細資訊，請參閱 IAM 使用者指南中的[身分 \(使用者、群組和角色\)](#)。

## 指定政策元素：動作、效果、資源和主體

針對每一個 Amazon ElastiCache 資源 (請參閱「[Amazon ElastiCache 資源和作業](#)」)，服務會定義一組 API 操作 (請參閱[動作](#))。為了授予這些 API 作業的許可，ElastiCache 定義了一組可在政策中指定的動作。例如，針對 ElastiCache 叢集資源定義的動作如下：CreateCacheCluster、DeleteCacheCluster 及 DescribeCacheCluster。執行一項 API 操作可能需要多個動作的許可。

以下是最基本的政策元素：

- 資源 – 在政策中，您可以使用 Amazon Resource Name (ARN) 來識別要套用政策的資源。如需更多詳細資訊，請參閱 [Amazon ElastiCache 資源和作業](#)。
- 動作：使用動作關鍵字識別您要允許或拒絕的資源操作。例如，根據指定的 Effect，elasticache:CreateCacheCluster 許可會允許或拒絕執行 Amazon ElastiCache CreateCacheCluster 作業的使用者許可。
- 效果 - 您可以指定使用者要求特定動作時會有什麼效果；可為允許或拒絕。如果您未明確授予存取 (允許) 資源，則隱含地拒絕存取。您也可以明確拒絕存取資源。例如，您可以這樣做以確保使用者無法存取資源，即使不同的政策授與存取。
- 主體：在以身分為基礎的政策 (IAM 政策) 中，政策所連接的使用者就是隱含主體。對於資源型政策，您可以指定想要收到許可的使用者、帳戶、服務或其他實體 (僅適用於資源型政策)。

如需進一步了解有關 IAM 政策語法和說明的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS IAM 政策參考](#)。

如需列出所有 Amazon ElastiCache API 動作的表格，請參閱「[ElastiCache API 權限：動作、資源和條件參考](#)」。

### 在政策中指定條件

當您授與許可時，您可以使用 IAM 政策語言指定政策生效時間的條件。例如，建議只在特定日期之後套用政策。如需使用政策語言指定條件的詳細資訊，請參閱 IAM 使用者指南中的 [條件](#)。

欲表示條件，您可以使用預先定義的條件金鑰。若要使用 ElastiCache 專用的條件索引鍵，請參閱「[使用條件索引鍵](#)」。您可以使用適當的 AWS 通用條件索引鍵。如需全 AWS 通用金鑰的清單，請參閱 IAM 使用者指南中的 [可用的條件金鑰](#)。

## Amazon ElastiCache 的 AWS 受管政策

AWS 管理的政策是由 AWS 建立和管理的獨立政策。AWS 管理的政策的設計在於為許多常見使用案例提供許可，如此您就可以開始將許可指派給使用者、群組和角色。

請謹記，AWS 管理的政策可能不會授予您特定使用案例的最低權限許可，因為它們可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的[客戶管理政策](#)，以便進一步減少許可。

您無法更改 AWS 管理的政策中定義的許可。如果 AWS 更新 AWS 管理的政策中定義的許可，更新會影響政策連接的所有主體身分 (使用者、群組和角色)。在推出新的 AWS 服務 或有新的 API 操作可供現有服務使用時，AWS 很可能會更新 AWS 管理的政策。

如需詳細資訊，請參閱 [《IAM 使用者指南》](#) 中的 AWS 受管政策。

### AWS 受管政策：ElastiCacheServiceRolePolicy

您無法將 ElastiCacheServiceRolePolicy 附加至 IAM 實體。此政策會附加至服務連結角色，而此角色允許 ElastiCache 代表您執行動作。

此政策允許 ElastiCache 代表您管理必要的 AWS 資源以管理快取：

- ec2 - 管理 EC2 網路資源以附加至快取節點，包括 VPC 端點 (適用無伺服器快取)、彈性網路介面 (ENI) (適用自行設計的叢集) 以及安全群組。
- cloudwatch - 將指標資料從服務發送到 CloudWatch。
- outposts - 允許在 AWS Outposts 上建立快取節點。

您可以在 IAM 主控台上找到 [ElastiCacheServiceRolePolicy](#) 政策，以及在 [《AWS 受管政策參考指南》](#) 中找到 [ElastiCacheServiceRolePolicy](#)。

### AWS 受管政策：AmazonElastiCacheFullAccess

您可將 AmazonElastiCacheFullAccess 政策連接到 IAM 身分。

此政策允許主體使用 AWS 管理主控台完整存取 ElastiCache：

- elasticache：存取所有 API。

- iam：建立服務運作所需的服務連結角色。
- ec2：說明建立快取所需的相依 EC2 資源 (VPC、子網路、安全群組)，並允許建立 VPC 端點 (適用於無伺服器快取)。
- kms：允許使用客戶受管 CMK 進行靜態加密。
- cloudwatch：允許存取指標以在主控台中顯示 ElastiCache 指標。
- application-autoscaling：允許存取以說明快取的自動擴展政策。
- logs：用於在主控台中填入日誌傳遞功能的日誌串流。
- firehose：用於在主控台中填入日誌傳遞功能的傳遞串流。
- s3：用於在主控台中填入快照還原功能的 S3 儲存貯體。
- outposts：用於在主控台中填入 AWS Outposts 以建立快取。
- sns：用於在主控台中填入通知功能的 SNS 主題。

您可以在 IAM 主控台上找到 [AmazonElastiCacheFullAccess](#) 政策，以及在《AWS 受管政策參考指南》中找到 [AmazonElastiCacheFullAccess](#)。

AWS 受管政策：AmazonElastiCacheReadOnlyAccess

您可將 AmazonElastiCacheReadOnlyAccess 政策連接到 IAM 身分。

此政策允許主體使用 AWS 管理主控台對 ElastiCache 進行唯讀存取：

- elasticache：存取唯讀 Describe API。

您可以在 IAM 主控台上找到 [AmazonElastiCacheReadOnlyAccess](#) 政策，以及在《AWS 受管政策參考指南》中找到 [AmazonElastiCacheReadOnlyAccess](#)。

AWS 受管政策的 ElastiCache 更新

檢視自此服務開始追蹤 ElastiCache AWS 受管政策更新以來的變更詳細資訊。如需有關此頁面變更的自動提醒，請訂閱 ElastiCache 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	日期
<a href="#">AmazonElastiCacheFullAccess</a> ：現有政策的更新	ElastiCache 新增了新許可，以允許管理無伺服器快取，並透過主控台啟用所有服務功能。	2023 年 11 月 27 日

變更	描述	日期
<a href="#">ElastiCacheServiceRolePolicy</a> : 現有政策的更新	ElastiCache 新增了新許可，以允許管理無伺服器快取資源的 VPC 端點。	2023 年 11 月 27 日
ElastiCache 已開始追蹤變更	ElastiCache 已開始追蹤其 AWS 受管政策的變更。	2020 年 2 月 07 日

## 針對 Amazon ElastiCache 使用身分型政策 (IAM 政策)

這個主題提供以身分為基礎的政策範例，在該政策中帳戶管理員可以將許可政策連接至 IAM 身分 (即使用者、群組和角色)。

### Important

建議您先閱讀說明管理 Amazon ElastiCache 資源存取基本概念與選項的主題。如需更多詳細資訊，請參閱 [管理 ElastiCache 資源存取許可的概觀](#)。

本主題中的各節涵蓋下列內容：

- [Amazon ElastiCache 的 AWS 受管政策](#)
- [客戶受管政策範例](#)

以下顯示許可政策範例。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowClusterPermissions",
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateServerlessCache",
      "elasticache:CreateCacheCluster",
      "elasticache:DescribeServerlessCaches",
      "elasticache:DescribeCacheClusters",
      "elasticache:ModifyServerlessCache",
      "elasticache:ModifyCacheCluster"
    ]
  }]
}
```

```
    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowUserToPassRole",
    "Effect": "Allow",
    "Action": [ "iam:PassRole" ],
    "Resource": "arn:aws:iam::123456789012:role/EC2-roles-for-cluster"
  }
]
```

此政策具有兩個陳述式：

- 第一個陳述式授予 Amazon ElastiCache 動作 (elasticache:Create\*、elasticache:Describe\*、elasticache:Modify\*) 的許可。
- 第二個陳述式會對 Resource 值結尾指定的 IAM 角色名稱授予 IAM 動作 (iam:PassRole) 的許可。

此政策不指定 Principal 元素，因為您不會在以身分為基礎的政策中，指定取得許可的主體。當您將政策連接至使用者時，這名使用者即為隱含主體。當您將許可政策連接至 IAM 角色，該角色的信任政策中所識別的委託人即取得許可。

如需詳列所有 Amazon ElastiCache API 動作及適用資源的表格，請參閱「[ElastiCache API 權限：動作、資源和條件參考](#)」。

## 客戶受管政策範例

如果您未使用預設政策並選擇使用自訂受管政策，請確保下列兩件事的其中一項。您應具有呼叫 iam:createServiceLinkedRole 的許可 (如需詳細資訊，請參閱[範例 4：允許使用者呼叫 IAM CreateServiceLinkedRole API](#))。或者，您應已建立 ElastiCache 服務連結角色。

與使用 Amazon ElastiCache 主控台所需的最低許可相結合時，本節中的範例政策會授予額外的許可。這些範例也與 AWS 軟體開發套件和 AWS CLI 相關。

如需設定 IAM 使用者和群組的說明，請參閱 IAM 使用者指南中的[建立您的第一個 IAM 使用者和管理員群組](#)。

### ⚠ Important

在生產環境中使用 IAM 政策之前，請一律先徹底進行測試。使用 ElastiCache 主控台時，一些看似簡單的 ElastiCache 動作可能需要其他動作來支援。例如，`elasticache:CreateCacheCluster` 可授予建立 ElastiCache 快取叢集的許可。但是，若要執行此操作，ElastiCache 主控台需使用許多 `Describe` 和 `List` 動作來填入主控台清單。

### 範例

- [範例 1：允許使用者對 ElastiCache 資源進行唯讀存取](#)
- [範例 2：允許使用者執行常見的 ElastiCache 系統管理員任務](#)
- [範例 3：允許使用者存取所有 ElastiCache API 動作](#)
- [範例 4：允許使用者呼叫 IAM `CreateServiceLinkedRole` API](#)

#### 範例 1：允許使用者對 ElastiCache 資源進行唯讀存取

下列政策會授予 ElastiCache 動作的許可，以允許使用者列出資源。您通常會將此類型的許可政策連接到管理員群組。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECReadOnly",
    "Effect": "Allow",
    "Action": [
      "elasticache:Describe*",
      "elasticache:List*"
    ],
    "Resource": "*"
  ]
}
```

#### 範例 2：允許使用者執行常見的 ElastiCache 系統管理員任務

常見的系統管理員任務包括修改資源。系統管理員也可能想要取得 ElastiCache 事件的相關資訊。下列政策為使用者授予執行 ElastiCache 動作的許可，以進行這些一般系統管理員任務。您通常會將此類型的許可政策連接到系統管理員群組。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECAAllowMutations",
    "Effect": "Allow",
    "Action": [
      "elasticache:Modify*",
      "elasticache:Describe*",
      "elasticache:ResetCacheParameterGroup"
    ],
    "Resource": "*"
  }
]
```

### 範例 3：允許使用者存取所有 ElastiCache API 動作

以下政策允許使用者存取所有 ElastiCache 動作。建議您只將此類型的許可政策授予管理員使用者。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECAAllowAll",
    "Effect": "Allow",
    "Action": [
      "elasticache:*"
    ],
    "Resource": "*"
  }
]
```

### 範例 4：允許使用者呼叫 IAM CreateServiceLinkedRole API

下列政策允許使用者呼叫 IAM CreateServiceLinkedRole API。建議您只將此類型的許可政策授予呼叫變動 ElastiCache 作業的使用者。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateSLRAllows",
```

```
    "Effect": "Allow",
    "Action": [
      "iam:CreateServiceLinkedRole"
    ],
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "elasticache.amazonaws.com"
      }
    }
  }
}
```

## 資源層級許可

您可以在 IAM 政策中指定資源來限制許可的範圍。許多 ElastiCache API 動作支援的資源類型取決於動作的行為。每個 IAM 政策陳述式授予在資源上執行動作的許可。當動作沒有作用於具名資源，或是當您授予對所有資源執行動作的許可，政策中資源的值是萬用字元 (\*)。對於許多 API 動作，您可以透過指定資源的 Amazon Resource Name (ARN) 或符合多個資源的 ARN 模式，來限制使用者可以修改的資源。若要依照資源限制許可，請依照 ARN 指定資源。

若要查看 ElastiCache 資源類型清單及其 ARN，請參閱《服務授權參考》中的 [Amazon ElastiCache 定義的資源](#)。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 [Amazon ElastiCache 定義的動作](#)。

### 範例

- [範例 1：允許使用者完整存取特定 ElastiCache 資源類型](#)
- [範例 2：拒絕使用者存取無伺服器快取。](#)

#### 範例 1：允許使用者完整存取特定 ElastiCache 資源類型

下列政策明確允許無伺服器快取類型的所有資源。

```
{
  "Sid": "Example1",
  "Effect": "Allow",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:*"
  ]
}
```

```
]
}
```

範例 2：拒絕使用者存取無伺服器快取。

下列範例會明確拒絕存取特定無伺服器快取。

```
{
  "Sid": "Example2",
  "Effect": "Deny",
  "Action": "elasticache:*",
  "Resource": [
    "arn:aws:elasticache:us-east-1:account-id:serverlesscache:name"
  ]
}
```

## 使用條件索引鍵

您可以指定條件，以決定 IAM 政策的生效方式。在 ElastiCache 中，您可以使用 JSON 政策的 Condition 元素來比較請求內容中的索引鍵和您在政策中指定的索引鍵值。如需詳細資訊，請參閱 [IAM JSON 政策元素：Condition](#)。

若要查看 ElastiCache 條件索引鍵的清單，請參閱《服務授權參考》中的 [Amazon ElastiCache 的條件索引鍵](#)。

如需全域條件索引鍵的清單，請參閱 [AWS 全域條件內容索引鍵](#)。

### 指定條件：使用條件金鑰

若要實作精細的控制機制，您可以撰寫 IAM 許可政策，指定條件來控制特定請求上的一組個別參數。您接著將政策套用至使用 IAM 主控台所建立的 IAM 使用者、群組或角色。

若要套用條件，請將條件資訊新增至 IAM 政策陳述式。在下列範例中，您會指定建立的任何自行設計快取叢集都將屬於節點類型 cache.r5.large 的條件。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
      "arn:aws:elasticache:*:*:parametergroup:*",
      "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
      "StringEquals": {
        "elasticache:CacheNodeType": [
          "cache.r5.large"
        ]
      }
    }
  }
]
}

```

如需詳細資訊，請參閱[標籤型存取控制政策範例](#)。

如需使用政策條件運算子的詳細資訊，請參閱「[ElastiCache API 權限：動作、資源和條件參考](#)」。

範例政策：使用條件進行精細參數控制

本節顯示對先前所列出的 ElastiCache 參數實作精細存取控制的範例政策。

1. `elasticache:MaximumDataStorage`：指定無伺服器快取的資料儲存上限。使用提供的條件，客戶就無法建立可儲存超過特定資料量的快取。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [

```

```

        "elasticache:CreateServerlessCache"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateServerlessCache"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
    ],
    "Condition": {
        "NumericLessThanEquals": {
            "elasticache:MaximumDataStorage": "30"
        },
        "StringEquals": {
            "elasticache:DataStorageUnit": "GB"
        }
    }
}
]
}

```

2. `elasticache:MaximumECPUPerSecond` : 指定無伺服器快取的每秒 ECPU 上限值。使用提供的條件，客戶就無法建立可執行超過每秒特定 ECPU 數量的快取。

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowDependentResources",
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateServerlessCache"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
                "arn:aws:elasticache:*:*:snapshot:*",

```

```

        "arn:aws:elasticache:*:*:usergroup:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateServerlessCache"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
    ],
    "Condition": {
        "NumericLessThanEquals": {
            "elasticache:MaximumECPUPerSecond": "100000"
        }
    }
}
]
}

```

3. `elasticache:CacheNodeType` : 指定使用者可以建立的節點類型。使用提供的條件，客戶可以指定節點類型的單一值或範圍值。

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [
                "arn:aws:elasticache:*:*:parametergroup:*",
                "arn:aws:elasticache:*:*:subnetgroup:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticache:CreateCacheCluster"
            ],
            "Resource": [

```

```

        "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
        "StringEquals": {
            "elasticache:CacheNodeType": [
                "cache.t2.micro",
                "cache.t2.medium"
            ]
        }
    }
}
]
}

```

#### 4. elasticache:EngineVersion : 指定使用 1.6.6 版引擎

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",
        "arn:aws:elasticache:*:*:subnetgroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:EngineVersion": "1.6.6"
        }
      }
    }
  ]
}

```

```

    ]
  }
}

```

5. `elasticache:KmsKeyId` : 指定使用客戶受管 AWS KMS 金鑰。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDependentResources",
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscachesnapshot:*",
        "arn:aws:elasticache:*:*:snapshot:*",
        "arn:aws:elasticache:*:*:usergroup:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateServerlessCache"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:serverlesscache:*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticache:KmsKeyId": "my-key"
        }
      }
    }
  ]
}

```

6. `elasticache:CacheParameterGroupName` : 使用叢集上組織的特定參數，指定非預設參數群組。您也可以為參數群組指定命名模式，或阻止刪除特定參數群組名稱。以下是限制僅使用「my-org-param-group」的範例。

```

{
  "Version": "2012-10-17",

```

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:parametergroup:*",
      "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
      "StringEquals": {
        "elasticache:CacheParameterGroupName": "my-org-param-group"
      }
    }
  }
]
}

```

7. `elasticache:CreateCacheCluster` : 如果請求標籤 `CreateCacheCluster` 遺失或不等於 `Project`、`Dev` 或 `QA` , 拒絕 `Prod` 動作。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticache:CreateCacheCluster"
      ],
      "Resource": [
        "arn:aws:elasticache:*:*:parametergroup:*",

```

```

        "arn:aws:elasticache:*:*:subnetgroup:*",
        "arn:aws:elasticache:*:*:securitygroup:*",
        "arn:aws:elasticache:*:*:replicationgroup:*"
    ],
},
{
    "Effect": "Deny",
    "Action": [
        "elasticache:CreateCacheCluster"
    ],
    "Resource": [
        "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
        "Null": {
            "aws:RequestTag/Project": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "elasticache:CreateCacheCluster",
        "elasticache:AddTagsToResource"
    ],
    "Resource": "arn:aws:elasticache:*:*:cluster:*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Project": [
                "Dev",
                "Prod",
                "QA"
            ]
        }
    }
}
]
}

```

8. `elasticache:CacheNodeType` : 允許 `CreateCacheCluster` 具有 `cacheNodeType` `cache.r5.large` 或 `cache.r6g.4xlarge` 以及 `Project=XYZ` 標籤。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:parametergroup:*",
      "arn:aws:elasticache:*:*:subnetgroup:*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "elasticache:CreateCacheCluster"
    ],
    "Resource": [
      "arn:aws:elasticache:*:*:cluster:*"
    ],
    "Condition": {
      "StringEqualsIfExists": {
        "elasticache:CacheNodeType": [
          "cache.r5.large",
          "cache.r6g.4xlarge"
        ]
      },
      "StringEquals": {
        "aws:RequestTag/Project": "XYZ"
      }
    }
  }
]
```

### Note

建立同時強制執行標籤和其他條件索引鍵的政策時，因為具有 `--tags` 參數的建立請求需要額外的 `elasticache:AddTagsToResource` 政策，`IfExists` 條件可能需要條件索引鍵元素。

## 為 Amazon ElastiCache 使用服務連結角色

Amazon ElastiCache 使用 AWS Identity and Access Management (IAM) [服務連結角色](#)。服務連結角色是一種獨特的 IAM 角色類型，可直接連結到 AWS 服務，例如 Amazon ElastiCache。Amazon ElastiCache 服務連結角色由 Amazon ElastiCache 預先定義。此角色包含服務需要的所有許可，以代您來呼叫其他的 AWS 服務。

服務連結角色可讓設定 Amazon ElastiCache 更為簡單，因為您不必手動新增必要的許可。角色已在 AWS 帳戶中，但會連結到 Amazon ElastiCache 使用案例，且已具備預先定義的許可。只有 Amazon ElastiCache 可以擔任這些角色，且只有這些角色可以使用預先定義的許可政策。您必須先刪除角色的相關資源，才能刪除角色。這可保護您的 Amazon ElastiCache 資源，避免您不小心移除存取資源所需的許可。

如需關於支援服務連結角色的其他服務的資訊，請參閱[可搭配 IAM 運作的 AWS 服務](#)，並尋找 Service-Linked Role (服務連結角色) 欄顯示為 Yes (是) 的服務。選擇具有連結的是，以檢視該服務的服務連結角色文件。

### 內容

- [Amazon ElastiCache 的服務連結角色許可](#)
  - [建立服務連結角色的許可](#)
- [建立服務連結角色 \(IAM\)](#)
  - [建立服務連結角色 \(IAM 主控台\)](#)
  - [建立服務連結角色 \(IAM CLI\)](#)
  - [建立服務連結角色 \(IAM API\)](#)
- [編輯 Amazon ElastiCache 服務連結角色的描述](#)
  - [編輯服務連結角色描述 \(IAM 主控台\)](#)
  - [編輯服務連結角色描述 \(IAM CLI\)](#)
  - [編輯服務連結角色描述 \(IAM API\)](#)
- [刪除 Amazon ElastiCache 的服務連結角色](#)
  - [清除服務連結角色](#)
  - [刪除服務連結角色 \(IAM 主控台\)](#)
  - [刪除服務連結角色 \(IAM CLI\)](#)
  - [刪除服務連結角色 \(IAM API\)](#)

## Amazon ElastiCache 的服務連結角色許可

### 建立服務連結角色的許可

#### 允取 IAM 實體建立 AWSServiceRoleForElastiCache 服務連結角色

將以下政策陳述式新增到該 IAM 實體的許可中：

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

#### 允取 IAM 實體刪除 AWSServiceRoleForElastiCache 服務連結角色

將以下政策陳述式新增到該 IAM 實體的許可中：

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

或者，您可以使用 AWS 受管政策，以提供 Amazon ElastiCache 的完整存取權。

### 建立服務連結角色 (IAM)

您可以使用 IAM 主控台、CLI 或 API 建立服務連結角色。

#### 建立服務連結角色 (IAM 主控台)

您可以使用 IAM 主控台建立服務連結角色。

## 建立服務連結角色 (主控台)

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)。然後選擇 Create new role (建立新角色)。
3. 在 Select type of trusted entity (選擇可信任的實體類型) 下，選擇 AWS Service (AWS 服務)。
4. 在 Or select a service to view its use cases (或者選取服務以檢視其使用案例) 下，選擇 ElastiCache。
5. 選擇 Next: Permissions (下一步：許可)。
6. 在 Policy name (政策名稱)，請注意 ElastiCacheServiceRolePolicy 是此角色的必要項目。選擇 Next: Add Tags (下一步：新增標籤)。
7. 請注意，服務連結的角色不支援標籤。選擇 Next: Review (下一步：檢閱)。
8. (選擇性) 針對 Role description (角色描述)，編輯新服務連結角色的描述。
9. 檢閱角色，然後選擇 Create role (建立角色)。

## 建立服務連結角色 (IAM CLI)

您可以從 AWS Command Line Interface 使用 IAM 作業來刪除服務連結角色。此角色可包含服務擔任該角色所需的信任政策與內嵌政策。

## 建立服務連結角色 (CLI)

使用以下操作：

```
$ aws iam create-service-linked-role --aws-service-name elasticache.amazonaws.com
```

## 建立服務連結角色 (IAM API)

您可以使用 IAM API 建立服務連結角色。此角色可包含服務擔任該角色所需的信任政策與內嵌政策。

## 建立服務連結角色 (API)

使用 [CreateServiceLinkedRole](#) API 呼叫。在請求中指定 `elasticache.amazonaws.com` 的服務名稱。

## 編輯 Amazon ElastiCache 服務連結角色的描述

Amazon ElastiCache 不允許您編輯 `AWSServiceRoleForElastiCache` service-linked 服務連結角色。因為有各種實體可能會參考服務連結角色，所以您無法在建立角色之後變更角色名稱。然而，您可使用 IAM 來編輯角色描述。

### 編輯服務連結角色描述 (IAM 主控台)

您可以使用 IAM 主控台來編輯服務連結角色描述。

### 編輯服務連結角色的說明 (主控台)

1. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)。
2. 選擇要修改之角色的名稱。
3. 在 Role description (角色說明) 的最右邊，選擇 Edit (編輯)。
4. 在方塊中輸入新的描述，然後選擇 Save (儲存)。

### 編輯服務連結角色描述 (IAM CLI)

您可以從 AWS Command Line Interface 使用 IAM 作業來編輯服務連結角色描述。

### 變更服務連結角色的說明 (CLI)

1. (選用) 若要檢視角色的目前描述，請使用 AWS CLI 執行 IAM 作業 [get-role](#)。

#### Example

```
$ aws iam get-role --role-name AWSServiceRoleForElastiCache
```

透過 CLI 操作，使用角色名稱 (而非 ARN) 來參照角色。例如，如果角色具有下列 ARN：`arn:aws:iam::123456789012:role/myrole`，請將角色參照為 **myrole**。

2. 若要更新服務連結角色的描述，請使用 AWS CLI 執行 IAM 作業 [update-role-description](#)。

若為 Linux、macOS 或 Unix：

```
$ aws iam update-role-description \  
  --role-name AWSServiceRoleForElastiCache \  
  --description "new description"
```

針對 Windows :

```
$ aws iam update-role-description ^  
  --role-name AWSServiceRoleForElastiCache ^  
  --description "new description"
```

編輯服務連結角色描述 (IAM API)

您可以使用 IAM API 來編輯服務連結角色描述。

變更服務連結角色的說明 (API)

1. (選用) 若要檢視角色的目前描述，請使用 IAM API 作業 [GetRole](#)。

Example

```
https://iam.amazonaws.com/  
  ?Action=GetRole  
  &RoleName=AWSServiceRoleForElastiCache  
  &Version=2010-05-08  
  &AUTHPARAMS
```

2. 若要更新角色的描述，請使用 IAM API 操作 [UpdateRoleDescription](#)。

Example

```
https://iam.amazonaws.com/  
  ?Action=UpdateRoleDescription  
  &RoleName=AWSServiceRoleForElastiCache  
  &Version=2010-05-08  
  &Description="New description"
```

刪除 Amazon ElastiCache 的服務連結角色

若您不再使用需要服務連結角色的功能或服務，我們建議您刪除該角色。如此一來，您就沒有未主動監控或維護的未使用實體。然而，務必清除您的服務連結角色，之後才能將其刪除。

Amazon ElastiCache 不會為您刪除服務連結角色。

## 清除服務連結角色

您必須先確認角色沒有與該角色相關聯的資源叢集，才能使用 IAM 來刪除服務連結角色。

檢查服務連結角色是否於 IAM 主控台有作用中的工作階段

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)。然後選擇 AWSServiceRoleForElastiCache 角色的名稱 (而非核取方塊)。
3. 在所選角色的 Summary (摘要) 頁面中，選擇 Access Advisor (存取 Advisor) 分頁。
4. 在 Access Advisor (存取 Advisor) 分頁中，檢閱服務連結角色的近期活動。

刪除需要 AWSServiceRoleForElastiCache 的 Amazon ElastiCache 資源

- 若要刪除叢集，請參閱下列指示：
  - [使用 AWS Management Console](#)
  - [使用 AWS CLI](#)
  - [使用 ElastiCache API](#)

刪除服務連結角色 (IAM 主控台)

您可以使用 IAM 主控台刪除服務連結角色。

刪除服務連結角色 (主控台)

1. 登入 AWS Management Console，並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇 Roles (角色)。然後，選擇您要刪除的角色名稱旁的核取方塊，而非名稱或資料列本身。
3. 在頁面頂端的 Role (角色) 動作中選擇 Delete (刪除) 角色。
4. 在確認對話方塊中，檢閱服務上次存取資料，以顯示每個所選取角色上次存取 AWS 服務的時間。這可協助您確認角色目前是否作用中。如果您想要繼續進行，請選擇 Yes, Delete (是，刪除) 來提交服務連結角色以進行刪除。

5. 查看 IAM 主控台通知，監視服務連結角色刪除的進度。因為 IAM 服務連結角色刪除不同步，所以在您提交角色進行刪除之後，刪除任務可能會成功或失敗。如果任務失敗，您可以從通知中選擇 View details (檢視詳細資訊) 或 View Resources (檢視資源)，以了解刪除失敗的原因。

### 刪除服務連結角色 (IAM CLI)

您可以從 AWS Command Line Interface 使用 IAM 作業來刪除服務連結角色。

### 刪除服務連結角色 (CLI)

1. 如果您不知道想要刪除的服務連結角色名稱，請輸入以下命令。此命令會列出您的帳戶中的角色及其 Amazon 資源名稱 (ARN)。

```
$ aws iam get-role --role-name role-name
```

透過 CLI 操作，使用角色名稱 (而非 ARN) 來參照角色。例如，如果角色的 ARN 為 `arn:aws:iam::123456789012:role/myrole`，參考這個角色時就需使用 **myrole**。

2. 因為無法刪除正在使用或具有相關聯資源的服務連結角色，所以您必須提交刪除要求。如果不符合這些條件，則可以拒絕該請求。您必須從回應中擷取 `deletion-task-id`，以檢查刪除任務的狀態。輸入下列內容，提交服務連結角色刪除請求。

```
$ aws iam delete-service-linked-role --role-name role-name
```

3. 輸入下列內容來檢查刪除任務的狀態。

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

刪除任務的狀態可以是 NOT\_STARTED、IN\_PROGRESS、SUCCEEDED 或 FAILED。如果刪除失敗，則呼叫會傳回失敗原因，以進行疑難排解。

### 刪除服務連結角色 (IAM API)

您可以使用 IAM API 刪除服務連結角色。

### 刪除服務連結角色 (API)

1. 若要提交服務連結名單的刪除請求，請呼叫 [DeleteServiceLinkedRole](#)。在請求中，指定角色名稱。

因為無法刪除正在使用或具有相關聯資源的服務連結角色，所以您必須提交刪除要求。如果不符合這些條件，則可以拒絕該請求。您必須從回應中擷取 `DeletionTaskId`，以檢查刪除任務的狀態。

2. 若要檢查刪除的狀態，請呼叫 [GetServiceLinkedRoleDeletionStatus](#)。在請求中，指定 `DeletionTaskId`。

刪除任務的狀態可以是 `NOT_STARTED`、`IN_PROGRESS`、`SUCCEEDED` 或 `FAILED`。如果刪除失敗，則呼叫會傳回失敗原因，以進行疑難排解。

## ElastiCache API 權限：動作、資源和條件參考

當您設定[存取控制](#)和寫入許可政策以附加到 IAM 政策 (以身分為基礎或以資源為基礎) 時，請使用下表作為參考。此表格列出每個 Amazon ElastiCache API 作業，以及您可授與執行動作權限的對應動作。您需在政策的 Action 欄位中指定動作，然後在政策的 Resource 欄位中指定資源值。除非另有說明，否則資源為必要項目。某些欄位同時包含必要資源和選用資源。如果沒有資源 ARN，則政策中的資源為萬用字元 (\*)。

您可以在 ElastiCache 原則中使用條件索引鍵來表示條件。若要查看 ElastiCache 特定條件索引鍵的清單，以及它們套用的動作和資源類型，請參閱[使用條件索引鍵](#)。如需全面金鑰的 AWS 完整清單，請參閱《IAM 使用者指南》中的[AWS 全域條件內容金鑰](#)。

### Note

若要指定動作，請使用後接 API 操作名稱的 elasticache: 字首 (例如，elasticache:DescribeCacheClusters)。

若要查看 ElastiCache 動作清單，請參閱服務授權參考 ElastiCache 中 [Amazon 定義的動作](#)。

## Amazon 的合規驗證 ElastiCache

協力廠商稽核人員會評估 AWS 服務的安全性與合規性，作為多重 AWS 合規計畫的一部分，例如 SOC、PCI、FedRAMP 和 HIPAA。

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱[AWS 服務 遵循規範計畫](#)方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱[AWS 規範計畫](#)。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱[下載中的報告中的 AWS Artifact](#)。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- [安全性與合規性快速入門指南](#) — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- [在 Amazon Web Services 上架構 HIPAA 安全性與合規性](#) — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 應用程式。

**Note**

並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 [HIPAA 資格服務參照](#)。

- [AWS 合規資源](#)[AWS](#) — 此工作簿和指南集合可能適用於您的產業和所在地。
- [AWS 客戶合規指南](#) — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- [使用 AWS Config 開發人員指南中的規則評估資源](#) — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- [AWS Security Hub](#) — 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 [Security Hub controls reference](#)。
- [Amazon GuardDuty](#) — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。GuardDuty 可協助您因應各種合規性需求，例如 PCI DSS，滿足特定合規性架構所規定的入侵偵測需求。
- [AWS Audit Manager](#) — 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

## 其他資訊

如需 AWS Cloud 合規性的一般資訊，請參閱下列內容：

- [依服務列出的 FIPS 端點](#)
- [服務更新 ElastiCache](#)
- [AWS 雲端合規](#)
- [共同的責任模型](#)
- [AWS PCI DSS 合規性計劃](#)

## Amazon ElastiCache 的備援功能

AWS 全球基礎設施是以 AWS 區域與可用區域為中心建置的。AWS 區域提供多個分開且隔離的實際可用區域，它們以低延遲、高輸送量和高度備援聯網功能相互連結。透過可用區域，您所設計與操作的應

應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需有關 AWS 區域與可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

除了 AWS 全球基礎設施外，Amazon ElastiCache 還提供數種支援資料復原和備份需求的功能。

主題

- [減少故障](#)

## 減少故障

規劃 Amazon ElastiCache 實作時，您應該進行規劃，使故障對應用程式和資料的影響降至最低。本節中的主題涵蓋您可以採取用來保護您的應用程式和資料不受故障的方法。

主題

- [緩解執行 Memcached 時的故障](#)
- [建議](#)

### 緩解執行 Memcached 時的故障

執行 Memcached 引擎時，您有下列選項可用於將故障的影響最小化。在您的故障緩解規劃中要解決兩個類型的故障：節點故障和可用區域故障。

#### 緩解節點故障

無伺服器快取會利用複寫的 Multi-AZ 架構自動緩解節點故障，讓應用程式清楚了解節點故障的情況。若要緩解自行設計叢集中節點故障的影響，請將您的快取資料分散到多個節點。因為自行設計的叢集不支援複寫，因此節點故障將一律造成您的叢集遺失一些資料。

當您創建 Memcached 集群時，您可以通過特殊請求創建 1 到 60 個節點或更多節點。將您的資料分割在更大量的節點間表示若節點故障，您遺失的資料較少。例如，如果將您的資料分割在 10 個節點間，任何單一節點大約會存放 10% 的快取資料。在此情況下，節點故障會遺失大約 10% 的快取，這個部分需要在建立和佈建替代節點時加以替代。如果在 3 個較大節點中快取相同的資料，節點的故障會遺失大約 33% 的快取資料。

如果您在 Memcached 叢集中需要 60 個以上的節點，或是一個 AWS 區域中的節點總計超過 300 個，請填寫 <https://aws.amazon.com/contact-us/elasticache-node-limit-request/> 的提高 ElastiCache 限制要求表單。

如需指定 Memcached 叢集中節點數量的詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。

## 緩解可用區域故障

無伺服器快取會利用複寫的 Multi-AZ 架構自動緩解可用區域故障，讓應用程式清楚了解 AZ 故障的情況。

若要緩解自行設計叢集中可用區域故障的影響，請盡可能將您的節點分散放到多個可用區域中。在極少數的 AZ 故障情況下，您會遺失在該 AZ 快取的資料，而非在其他 AZ 快取的資料。

### 為什麼要有這麼多節點？

如果我的區域只有 3 個可用區域，既然若一個 AZ 故障，我會遺失大約 1/3 的資料，為什麼我需要超過 3 個節點？

這是個好問題。請記得我們正嘗試緩解兩個獨特類型的故障，節點和可用區域。您是對的，如果您的資料分散在可用區域間，當其中一個區域故障時，您只會遺失在該 AZ 中快取的資料，而不論您擁有的節點數量為何。不過，如果某個節點故障，擁有的節點數量越多可減少資料遺失的比例。

沒有「神奇公式」可決定您叢集中的節點數量。您必須衡量資料遺失的影響、故障可能性與成本，並從中得出自己的結論。

如需指定 Memcached 叢集中節點數量的詳細資訊，請參閱[建立 Memcached 叢集 \(主控台\)](#)。

如需區域和可用區域的詳細資訊，請參閱[區域和可用區域](#)。

## 建議

我們建議您在自行設計的叢集上建立無伺服器快取，如此您無需額外的組態設定就能自動獲得更優異的容錯能力。不過，建立自行設計的叢集時，您需要針對兩個類型的故障進行規劃：個別節點故障和廣泛的可用區域故障。最佳的故障緩解規劃可解決這兩種類型的故障。

### 將節點故障的影響降至最低

執行 Memcached 並在節點間分割您的資料時，使用的節點愈多，任何節點故障時遺失的資料愈少。

### 將可用區域故障的影響降到最低

若要將可用區域故障的影響降到最低，建議您在盡可能多的不同可用區域中啟動節點。在可用區域間平均分散節點，可將很少見的可用區域故障影響降到最低。無伺服器快取會自動進行此操作。

# AWS ElastiCache 中的基礎設施安全性

AWS ElastiCache 為受管服務，受到 AWS 全域網路安全性程序的保護，如 [AWS 架構中心](#) 的「安全性與合規」一節中所述。

您可使用 AWS 發佈的 API 呼叫，透過網路來存取 ElastiCache。用戶端必須支援 Transport Layer Security (TLS) 1.2 或更新版本。建議使用 TLS 1.3 或更新版本。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 以產生暫時安全憑證以簽署請求。

## 服務更新 ElastiCache

ElastiCache 自動監控您的快取、叢集和節點叢集，以便在服務更新可用時套用服務更新。無伺服器快取的服務更新會自動且公開套用。對於自行設計的叢集，您可以設定預先定義的維護時段，ElastiCache 以套用這些更新。不過，在某些情況下，您可能會發現此方法過於嚴格，而且可能會限制您的業務流程。

您可以使用服務更新來控制要對您的自行設計叢集套用哪些更新，以及套用的時間。您也可以即時監控所選 ElastiCache 叢集的這些更新進度。

## 管理服務更新

ElastiCache 會定期發行自行設計叢集的服務更新。如果您有一或多個適用於這些服務更新的自行設計叢集，則會在更新發佈時透過電子郵件、SNS、Personal Health Dashboard (PHD) 和 Amazon CloudWatch 事件收到通知。更新也會顯示在 ElastiCache 主控台的 [服務更新] 頁面上。透過使用此儀表板，您可以檢視 ElastiCache 叢集的所有服務更新及其狀態。無伺服器快取的服務更新會公開套用，且無法透過服務更新管理。

您可以在自動更新開始前，控制套用更新的時間。強烈建議您盡快套用安全性更新類型的任何更新，以確保 ElastiCache 叢集始終 up-to-date 具有最新的安全性修補程式。

以下區段更會詳細探討這些選項。

### 主題

- [套用服務更新](#)

- [確認您已使用 AWS 主控台套用最新的服務更新](#)
- [停止服務更新](#)

## 套用服務更新

您可以在更新的狀態變成 available (可用) 時，開始將服務更新套用到機群。服務更新為累積更新。換句話說，您未套用的任何更新都會包含在最新的更新中。

如果服務更新已啟用自動更新，您可以選擇在可用時不採取任何動作。ElastiCache 將排程在自動更新開始日期之後，在叢集即將進行的其中一個維護時段中套用更新。您將收到更新每個階段的相關通知。

### Note

您只能套用狀態為 available (可用) 或 scheduled (已排程) 的服務更新。

如需有關檢閱任何服務特定更新並套用至適用 ElastiCache 叢集的詳細資訊，請參閱[使用主控台套用服務更新](#)。

當一或多個 ElastiCache 叢集有新的服務更新可用時，您可以使用 ElastiCache 主控台、API 或套 AWS CLI 用更新。以下區段會說明您可以用來套用更新的選項。

### 使用主控台套用服務更新

若要檢視可用服務更新的清單與其他資訊，請前往主控台的 Service Updates (服務更新) 頁面。

1. 登錄到 AWS Management Console 並打開 Amazon ElastiCache 控制台 <https://console.aws.amazon.com/elasticache/>。
2. 在導覽窗格上，選擇 Service Updates (服務更新)。
3. 在 Service updates (服務更新) 中，您可以檢視下列項目：
  - Service update name (服務更新名稱)：服務更新的唯一名稱
  - Update type (更新類型)：服務更新的類型，可為 security-update (安全性更新) 或 engine-update (引擎更新)
  - Update severity (更新嚴重性)：套用更新的優先順序：
    - critical (重大)：建議您立即套用此更新 (在 14 天內)。
    - important (重要)：建議您盡快在業務流程允許的情況下套用此更新 (在 30 天內)。
    - medium (中等)：建議您盡快套用此更新 (在 60 天內)。

- low (低度)：建議您盡快套用此更新程式 (在 90 天內)。
  - Engine version (引擎版本)：如果更新類型為 engine-update (引擎更新)，表示正在更新的引擎版本。
  - 發佈日期：更新發佈且可在叢集上套用的時間。
  - 建議套用日期：套用更新的 ElastiCache 指引日期。
  - 狀態：更新的狀態，可以是下列狀態之一：
    - 可用：更新可用於必要的叢集。
    - complete (完成)：已套用更新。
    - cancelled (已取消)：更新已取消且不再需要。
    - 過期：更新已不可套用。
4. 選擇個別更新 (而非其左側的按鈕)，檢視服務更新的詳細資訊。

在 Cluster update status (叢集更新狀態) 區段中，您可以檢視叢集清單，這些叢集尚未套用或最近才套用服務更新。針對每個叢集，您可以檢視下列項目：

- Cluster name (叢集名稱)：叢集的名稱
- Nodes updated (已更新的節點)：特定叢集中的個別節點比率，這些節點已更新或仍可用於特定服務更新。
- Update Type (更新類型)：服務更新的類型，可為 security-update (安全性更新) 或 engine-update (引擎更新)
- Status (狀態)：叢集上服務更新的狀態，可以是下列狀態之一：
  - available (可用)：更新可用於必要的叢集。
  - 進行中：正在將更新套用至此叢集。
  - scheduled (已排程)：已排程更新日期。
  - complete (完成)：已成功套用更新。狀態為 complete (完成) 的叢集會在完成後顯示 7 天。

如果您已選擇任何或所有狀態為 available (可用) 或 scheduled (已排程) 的叢集，然後選擇 Apply now (立即套用)，系統會開始將這些更新套用至這些叢集。

## 使用 AWS CLI 套用服務更新

在收到服務更新可供使用的通知後，您就可以使用 AWS CLI 來檢查和套用這些更新：

- 若要擷取可用服務更新的說明，請執行下列命令：

```
aws elasticache describe-service-updates --service-update-status
available
```

如需詳細資訊，請參閱[describe-service-updates](#)。

- 若要在叢集清單上套用服務更新，請執行下列命令：

```
aws elasticache batch-apply-update-action --service-update
ServiceUpdateNameToApply=sample-service-update --cluster-names cluster-1
cluster2
```

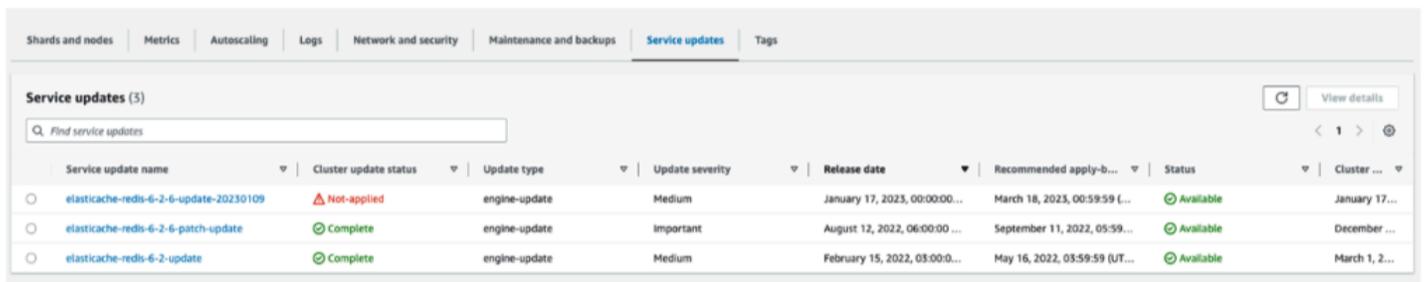
如需詳細資訊，請參閱[batch-apply-update-action](#)。

## 確認您已使用 AWS 主控台套用最新的服務更新

您可以按照以下步驟驗證您的 ElastiCache Redis 叢集是否正在執行最新的服務更新：

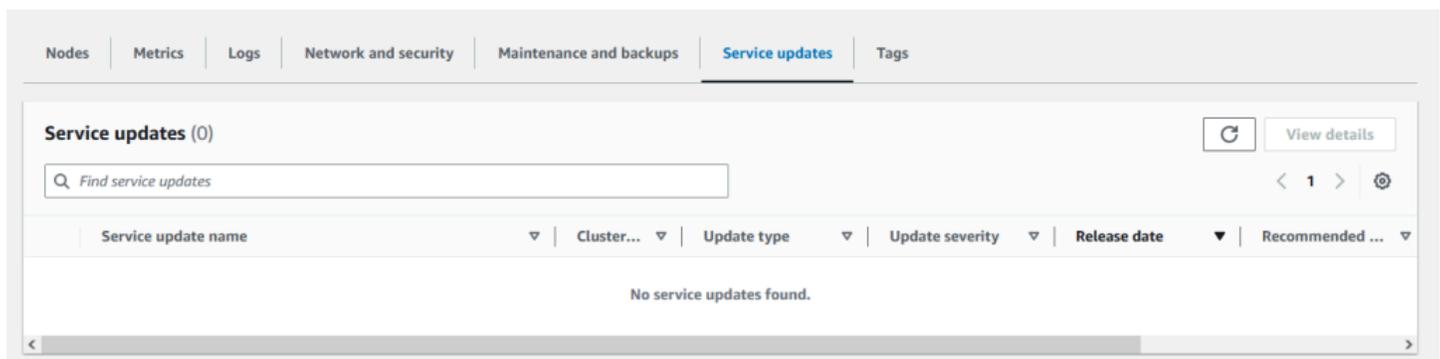
1. 在 Redis 叢集頁面上選擇適用的叢集
2. 在瀏覽窗格中選擇 [服務更新]，以查看該叢集適用的服務更新 (如果有的話)。

如果主控台顯示服務更新清單，您可以選取服務更新並選擇 [立即套用]。



Service update name	Cluster update status	Update type	Update severity	Release date	Recommended apply-b...	Status	Cluster ...
elasticache-redis-6-2-6-update-2023109	Not-applied	engine-update	Medium	January 17, 2023, 00:00:00...	March 18, 2023, 00:59:59 (...)	Available	January 17...
elasticache-redis-6-2-6-patch-update	Complete	engine-update	Important	August 12, 2022, 06:00:00 ...	September 11, 2022, 05:59...	Available	December ...
elasticache-redis-6-2-update	Complete	engine-update	Medium	February 15, 2022, 03:00:0...	May 16, 2022, 05:59:59 (J/T...	Available	March 1, 2...

如果主控台顯示「找不到服務更新」，表示 ElastiCache 針對 Redis 叢集已套用最新的服務更新。



Service update name	Cluster...	Update type	Update severity	Release date	Recommended ...
No service updates found.					

## 停止服務更新

如有需要，您可以停止叢集更新。例如，如果正在進行更新的叢集發生非預期的激增，建議您停止更新。或者，如果更新花費太長時間，而且在尖峰時間中斷業務流程，您可能會想要停止更新。

[停止中](#)操作會立即中斷對那些叢集以及還未更新之任何節點的所有更新。此操作會繼續完成狀態為 in progress (處理中) 的任何節點。然而，它會停止對相同叢集中狀態為 update available (可更新) 的其他節點的更新，並將這些節點還原成 Stopping (停止中) 狀態。

Stopping (停止中) 工作流程已完成，狀態為 Stopping (停止中) 的節點會變更為 Stopped (已停止) 狀態。根據更新的工作流程，某些叢集不會更新任何節點。其他叢集可能會包含一些已更新的節點以及狀態仍是 update available (可更新) 的節點。

您可以稍後返回，在業務流程允許的時候再完成更新進度。在這種情況下，請選擇您想要完成更新的適用叢集，接著選擇 Apply Now (立即套用)。如需詳細資訊，請參閱 [套用服務更新](#)。

### 使用主控台

您可以使用 ElastiCache 主控台中斷服務更新。以下內容會示範其做法：

- 在選取的叢集上進行服務更新之後，ElastiCache 主控台會在儀表板頂端顯示 [檢視/停止更新] 索引標籤。ElastiCache
- 若要中斷更新，請選擇 Stop Update (停止更新)。
- 停止更新時，選擇叢集並檢查狀態。叢集會還原成 Stopping (停止中) 狀態，最後會變成 Stopped (已停止) 狀態。

### 使用 AWS CLI

您可以使用 AWS CLI 來中斷服務更新。下列程式碼範例示範其做法：

若是複寫群組，請執行下列操作：

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --replication-group-ids my-replication-group-1 my-replication-group-2
```

若是快取叢集，請執行下列操作：

```
aws elasticache batch-stop-update-action --service-update-name sample-service-update --cache-cluster-ids my-cache-cluster-1 my-cache-cluster-2
```

如需詳細資訊，請參閱[BatchStopUpdateAction](#)。

# 在 Amazon ElastiCache 中記錄和監控

若要管理快取，您務必要了解快取的執行方式。ElastiCache 會產生發佈至 Amazon CloudWatch Logs 的指標，用於監控快取效能。此外，ElastiCache 會在快取資源發生重大變更時產生事件 (例如，建立新快取或刪除快取)。

## 主題

- [無伺服器指標和事件](#)
- [自行設計叢集的指標和事件](#)
- [使用 AWS CloudTrail 記錄 Amazon ElastiCache API 呼叫](#)
- [使用 AWS CloudTrail 記錄 Amazon ElastiCache API 呼叫](#)

## 無伺服器指標和事件

本節說明使用無伺服器快取時可監控的指標和事件。

## 主題

- [無伺服器快取指標](#)
- [無伺服器快取事件](#)

## 無伺服器快取指標

AWS/ElastiCache 命名空間包含適用於 Redis 無伺服器快取的下列 CloudWatch 指標。

指標	描述	單位
BytesUsedForCache	儲存在快取中的資料所使用的位元組總數。	位元組
ElastiCacheProcessingUnits	在快取上執行的請求所消耗的 ElastiCacheProcessingUnits (ECPU) 總數	計數

指標	描述	單位
SuccessfulReadRequestLatency	成功讀取請求的延遲。	微秒
SuccessfulWriteRequestLatency	成功寫入請求的延遲	微秒
TotalCmdsCount	快取上執行的所有命令總數	計數
CurrConnections	快取的用戶端連線數。	計數
ThrottledCmds	因為工作負載的擴展速度比 ElastiCache 能夠擴展的速度快，而遭到 ElastiCache 限流的請求數。	計數
NewConnections	在此期間內，伺服器已接受的連線總數。	計數
CurrItems	快取中的項目數。	計數
NetworkBytesIn	傳入快取的位元組總數	位元組
NetworkBytesOut	從快取傳出的位元組總數	位元組
移出	由快取移出的金鑰計數	計數
已回收	快取使其到期的金鑰數目	計數

## 命令層級指標

ElastiCache 也會發出下列 Memcached 命令層級指標

指標	描述	單位
CmdGet	快取已接收到的 get 命令的數量。	計數

指標	描述	單位
CmdSet	快取接收到的 set 命令數量。	計數
CmdTouch	快取已接收到的 touch 命令數量。	計數
GetHits	在請求的金鑰已找到的情況下，快取接收到的取得請求的數量。	計數
GetMisses	快取已接收到其請求的金鑰未找到的取得請求的數量。	計數
IncrHits	在請求的金鑰已找到的情況下，快取接收到的增加請求的數量。	計數
IncrMisses	在請求的金鑰未找到的情況下，快取接收到的增加請求的數量。	計數
DecrHits	在請求的金鑰已找到的情況下，快取接收到的減少請求的數量。	計數
DecrMisses	在請求的金鑰未找到的情況下，快取接收到的減少請求的數量。	計數
DeleteHits	在請求的金鑰已找到的情況下，快取接收到的刪除請求的數量。	計數
DeleteMisses	在請求的金鑰未找到的情況下，快取接收到的刪除請求的數量。	計數

指標	描述	單位
TouchHits	已被碰觸並賦予新的過期時間的金鑰的數量。	計數
TouchMisses	已執行 touch 但未找到的金鑰數量。	計數
CasHits	在請求的金鑰已找到且 cas 值相符的情況下，快取已接收到的 cas 請求數量。	計數
CasMisses	在請求的金鑰未找到的情況下，快取已接收到 cas 請求數量。	計數
CasBadval	在 cas 值與儲存的 cas 值不相符的情況下，快取已接收到的 cas 請求數量。	計數
CmdFlush	快取已接收到的 flush 命令的數量。	計數

## 無伺服器快取事件

與您的無伺服器快取相關的 ElastiCache 日誌事件。此資訊包含事件的日期和時間、事件的來源名稱和來源類型，以及事件的描述。您可以使用 ElastiCache 主控台、AWS CLI describe-events 命令或 ElastiCache API 動作 DescribeEvents，輕鬆從日誌擷取事件。

您可以選擇使用 Amazon EventBridge 來監控、擷取、轉換和處理 ElastiCache 事件。在 Amazon EventBridge <https://docs.aws.amazon.com/eventbridge/latest/userguide/> 中進一步了解。

### 檢視 ElastiCache 事件 (主控台)

使用 ElastiCache 主控台檢視事件

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。

- 若要查看所有可用事件的清單，請在導覽窗格中選擇 Events (事件)。
- 在事件畫面上，清單中的每一列都代表一個事件，並顯示事件來源、事件類型、事件的 GMT 時間以及事件的說明。您可以使用 Filter (篩選條件) 指定要查看事件清單中的所有事件，還是只查看特定類型的事件。

## 檢視 ElastiCache 事件 (AWS CLI)

若要使用 AWS CLI 產生 ElastiCache 事件的清單，請使用 `describe-events` 命令。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼最多會列出 40 個無伺服器快取事件。

```
aws elasticache describe-events --source-type serverless-cache --max-items 40
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內無伺服器快取的所有事件。

```
aws elasticache describe-events --source-type serverless-cache --duration 1440
```

## 無伺服器事件

本節記錄您可能收到的不同類型無伺服器快取事件。

### 無伺服器快取建立事件

詳細資訊類型	描述	單位	來源	訊息
快取已建立	Cache arn	建立	serverless-cache	快取 <cache-name> 已建立且可供使用。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。沒有足夠的可用 IP 地址可用來建立 VPC 端點。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。

詳細資訊類型	描述	單位	來源	訊息
				請求中提供的子網路無效。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。已達建立 VPC 端點的配額限制。
快取建立失敗	Cache arn	失敗	serverless-cache	無法建立快取 <cache-name>。您沒有建立 VPC 端點的許可。

### 無伺服器快取更新事件

詳細資訊類型	資源清單	類別	來源	訊息
快取已更新	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的 SecurityGroups 已更新。
快取已更新	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的標籤已更新。
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 更新失敗。SecurityGroups 更新失敗。
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 更新失敗。SecurityGroups 更新

詳細資訊類型	資源清單	類別	來源	訊息
				失敗，因為許可不足。
快取更新失敗	Cache arn	組態變更	serverless-cache	快取 <cache-name> 更新失敗。SecurityGroups 更新失敗，因為 SecurityGroups 無效。

#### 無伺服器快取刪除事件

詳細資訊類型	資源清單	類別	來源	訊息
快取已刪除	Cache arn	刪除	serverless-cache	快取 <cache-name> 已刪除。

#### 無伺服器快取用量限制事件

詳細資訊類型	描述	單位	來源	訊息
快取已更新	Cache arn	組態變更	serverless-cache	快取 <cache-name> 的限制已更新。
快取更新失敗	Cache arn	失敗	serverless-cache	快取 <cache-name> 的限制更新失敗，因為快取已刪除。
快取更新失敗	Cache arn	失敗	serverless-cache	快取 <cache-name> 的限制更新

詳細資訊類型	描述	單位	來源	訊息
				失敗，因為組態無效。

## 自行設計叢集的指標和事件

本節說明使用自行設計的叢集時，預期會看到的指標、事件和日誌。

### 主題

- [自行設計叢集的指標](#)
- [自行設計叢集的事件](#)
- [使用 CloudWatch 指標監控用量](#)
- [Amazon SNS 的 ElastiCache 事件監控](#)

## 自行設計叢集的指標

當您自行設計叢集時，ElastiCache 會在每個節點層級發出指標，包括主機層級指標和快取指標。

如需 Memcached 的主機層級指標的詳細資訊，請參閱 [主機層級指標](#)。

如需節點層級 Memcached 指標的詳細資訊，請參閱 [Memcached 的指標](#)。

## 自行設計叢集的事件

與您的自行設計快取相關的 ElastiCache 日誌事件。使用自行設計的叢集時，您可以在 ElastiCache 主控台中、使用 AWS CLI 或使用 Amazon Simple Notification Service (SNS) 檢視叢集事件。自行設計的叢集事件不會發佈至 Amazon EventBridge。

自行設計的叢集事件資訊包含事件的日期和時間、事件的來源名稱和來源類型，以及事件的說明。您可以使用 ElastiCache 主控台、AWS CLI describe-events 命令或 ElastiCache API 動作 DescribeEvents，輕鬆從日誌擷取事件。

### 檢視 ElastiCache 事件 (主控台)

下列程序使用 ElastiCache 主控台顯示事件。

### 使用 ElastiCache 主控台檢視事件

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用事件的清單，請在導覽窗格中選擇 Events (事件)。
3. 在事件畫面上，清單中的每一列都代表一個事件，並顯示事件來源、事件類型、事件的 GMT 時間以及事件的說明。您可以使用 Filter (篩選條件) 指定要查看事件清單中的所有事件，還是只查看特定類型的事件。

## 檢視 ElastiCache 事件 (AWS CLI)

若要使用 AWS CLI 產生 ElastiCache 事件的清單，請使用 `describe-events` 命令。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼最多會列出 40 個自行設計叢集事件。

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內自行設計快取的所有事件。

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

## 自行設計的叢集事件

本節包含您預期會收到的自行設計叢集事件清單。

下列 ElastiCache 事件會觸發 Amazon SNS 通知。如需事件的詳細資訊，請參閱 [檢視 ElastiCache 事件](#)。

事件名稱	訊息	描述
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	快取節點已新增至快取叢集，且已就緒可供使用。
ElastiCache:AddCacheNodeFailed，因可用的 IP 地址不足所導致	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	由於可用的 IP 地址不足，而無法新增快取節點。

事件名稱	訊息	描述
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	一或多個快取叢集參數已變更。
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	快取叢集的佈建已完成，且快取叢集中的快取節點已就緒並可供使用。
ElastiCache:CacheClusterProvisioningFailed，因不相容的網路狀態所導致	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	已嘗試將新快取叢集啟動至不存在的虛擬私有雲端 (VPC)。
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	快取叢集的擴展已成功完成。
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed: <i>cluster-name</i>	快取叢集上的擴展操作失敗。
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	已發生下列其中一項事件： <ul style="list-style-type: none"> <li>• 授權給快取叢集的快取安全群組清單已修改。</li> <li>• 一或多個新 EC2 安全群組，已在與快取叢集相關聯的任何快取安全群組上授權。</li> <li>• 一或多個新 EC2 安全群組，已從與快取叢集相關聯的任何快取安全群組上撤銷。</li> </ul>

事件名稱	訊息	描述
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機已降級或無法連線，且已開始取代快取節點。</p> <div data-bbox="1068 401 1507 617" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>取代之快取節點的 DNS 項目並未變更。</p></div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。然而，某些快取用戶端程式庫可能會在 ElastiCache 取代快取節點後停止使用快取節點；在這種情況下，應用程式應在此事件發生時重新整理伺服器清單。</p>

事件名稱	訊息	描述
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機已降級或無法連線，且已完成取代快取節點。</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>取代之快取節點的 DNS 項目並未變更。</p> </div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。然而，某些快取用戶端程式庫可能會在 ElastiCache 取代快取節點後停止使用快取節點；在這種情況下，應用程式應在此事件發生時重新整理伺服器清單。</p>
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>一或多個快取節點已重新開機。</p> <p>訊息 (Memcached) : "Cache node %s shutdown" 然後，第二個訊息 : "Cache node %s restarted"</p>
ElastiCache:CertificateRenewalComplete (僅 Redis)	ElastiCache:CertificateRenewalComplete	Amazon CA 憑證已成功續約。
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	已成功建立複寫群組。

事件名稱	訊息	描述
ElastiCache:DeleteCacheClusterComplete	ElastiCache:DeleteCacheClusterComplete : <i>cluster-name</i>	已完成刪除快取叢集和所有與其相關聯的快取節點。
ElastiCache:FailoverComplete (僅 Redis)	ElastiCache:FailoverComplete : <i>mycluster</i>	已成功容錯移轉至複本節點。
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	叢集中的複本數量已增加。
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	將複本新增至叢集的程序已經開始。
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	叢集中原先已排程替換的節點，已不再排程替換。
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>叢集中原先已排程替換的節點，已重新排程在通知中所述的新視窗期間進行替換。</p> <p>如需您可以採取哪些動作的相關資訊，請參閱<a href="#">取代 Memcached 的快取節點</a>。</p>

事件名稱	訊息	描述
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	您叢集中的節點，已排在通知中所述的視窗期間進行替換。  如需您可以採取哪些動作的相關資訊，請參閱 <a href="#">取代 Memcached 的快取節點</a> 。
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	快取節點已從快取叢集移除。
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	已順利完成複寫群組上的向上擴展操作。
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	複寫群組上的擴展操作失敗。
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	節點有可用的自助更新。
ElastiCache:SnapshotComplete (僅 Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	快取快照已順利完成。
ElastiCache:SnapshotFailed (僅 Redis)	SnapshotFailed : <i>cluster-name</i>	快取快照已失敗。請參閱叢集的快取事件，以取得失敗原因的詳細資訊。  如果您描述快照，請查看 <a href="#">DescribeSnapshots</a> ，其狀態將是 failed。

## 使用 CloudWatch 指標監控用量

ElastiCache 提供指標，可讓您監控您的叢集。您可以透過 CloudWatch 存取這些指標。如需 CloudWatch 的詳細資訊，請參閱 [CloudWatch 說明文件](#)。

ElastiCache 提供主機層級指標 (例如 CPU 用量) 及快取引擎軟體的特定指標 (例如快取取得及快取遺失)。每隔 60 秒會針對每個快取節點測量及發佈這些指標。

### Important

您應考慮在某些關鍵指標設定 CloudWatch 警示，讓您可以在快取叢集的效能開始降低時接收通知。如需詳細資訊，請參閱本指南中的 [我應該監控哪些指標？](#)。

### 主題

- [主機層級指標](#)
- [Memcached 的指標](#)
- [我應該監控哪些指標？](#)
- [監控 CloudWatch 叢集和節點指標](#)

### 主機層級指標

AWS/ElastiCache 命名空間包含下列個別快取節點的主機層級指標。

另請參閱

- [Memcached 的指標](#)

指標	描述	單位
CPUUtilization	整部主機 CPU 使用率百分比。	百分比
CPUCreditBalance	自執行個體啟動或開始後，累積獲得的 CPU 額度數量。如果是 T2 Standard，CPUCreditBalance 也包含已產生的啟動額度。  獲得額度後，額度會在額度餘額中累積，並在支付額度時，從額度餘額中移出。額度餘額有最大	額度 (vCPU-分鐘)

指標	描述	單位
	<p>值限制，它取決於執行個體大小。到達限制之後，任何獲得的新額度都會遭到捨棄。如果是 T2 Standard，啟動額度不會計入此限制。</p> <p>CPUCreditBalance 中的額度可供執行個體支付以大幅提升並超越基準 CPU 使用率。</p> <p>CPU 額度指標僅提供 5 分鐘頻率。</p>	
CPUCreditUsage	<p>執行個體為 CPU 使用率花費的 CPU 額度數量。一個 CPU 額度等於一個 vCPU 以 100% 使用率執行 1 分鐘，或同等的 vCPU、使用率與時間的組合 (例如，一個 vCPU 以 50% 使用率執行 2 分鐘，或兩個 vCPU 以 25% 使用率執行 2 分鐘)。</p> <p>CPU 額度指標僅提供 5 分鐘頻率。如果您要指定大於 5 分鐘的期間，請使用總和統計數字代替平均統計數字。</p>	額度 (vCPU-分鐘)
FreeableMemory	主機上可用的記憶體數量。這來自於 OS 報告為可用的 RAM、緩衝區和快取。	位元組
NetworkBytesIn	主機已從網路讀取的位元組數。	位元組
NetworkBytesOut	執行個體在所有網路界面上送出的位元組數目。	位元組
NetworkPacketsIn	執行個體在所有網路界面上收到的封包數目。此指標識別單一執行個體上的傳入流量 (封包數目)。	計數
NetworkPacketsOut	執行個體在所有網路界面上送出的封包數目。此指標識別單一執行個體上的傳出流量 (封包數目)。	計數
NetworkBandwidthIn AllowanceExceeded	因傳入的彙總頻寬超過執行個體的上限而成形的封包數目。	計數

指標	描述	單位
NetworkConntrackAllowanceExceeded	因為連線追蹤超過執行個體的上限且無法建立新的連線，而成形的封包數目。這可能會導致傳送或傳回執行個體流量的封包遺失。	計數
NetworkBandwidthOutAllowanceExceeded	因傳出的彙總頻寬超過執行個體的上限而成形的封包數目。	計數
Network Packets Per Second Allowance Exceeded	因雙向每秒封包數超過執行個體的上限而成形的封包數目。	計數
NetworkMaxBytesIn	每分鐘內接收到的最大位元組高載。	位元組
NetworkMaxBytesOut	每分鐘內傳輸的最大位元組高載。	位元組
NetworkMaxPacketsIn	每分鐘內接收到的最大封包高載。	計數
NetworkMaxPacketsOut	每分鐘內傳輸的最大封包高載。	計數
SwapUsage	主機已使用的交換空間的量。	位元組

## Memcached 的指標

AWS/ElastiCache 命名空間包含下列 Memcached 指標。

AWS/ElastiCache 命名空間包含以下衍生自 Memcached 統計資料命令的度量。每個指標都是在快取模式層級進行計算。

另請參閱

- [主機層級指標](#)

指標	描述	單位
BytesReadIntoMemcached	快取節點從網路讀取的位元組數。	位元組

指標	描述	單位
BytesUsedForCacheItems	用於存放快取項目的位元組數。	位元組
BytesWrittenOutFromMemcached	快取節點寫入至網路的位元組數。	位元組
CasBadval	快取已接收到且 Cas 值與已儲存的 Cas 值不匹配的 CAS (檢查與設定) 請求的數量。	計數
CasHits	快取已接收到，其請求的金鑰已找到且 Cas 值匹配的 Cas 請求的數量。	計數
CasMisses	快取已接收到其請求的金鑰未找到的 Cas 請求的數量。	計數
CmdFlush	快取已接收到的 flush 命令的數量。	計數
CmdGet	快取已接收到的 get 命令的數量。	計數
CmdSet	快取接收到的 set 命令數量。	計數
CurrConnections	<p>即時連接至快取的連接數量計數。ElastiCache 使用兩到三個連線來監視叢集。</p> <p>除了以上之外，memcached 會建立一些內部連線，等於用於節點類型的兩倍執行緒。各種節點類型的執行緒計數，可於 Nodetype Specific Parameters 適用的參數群組中查看。</p> <p>連線總數是用戶端連線、監控的連線和以上提及的內部連線的總和。</p>	計數
CurrItems	目前儲存於快取的項目數量計數。	計數
DecrHits	在請求的金鑰已找到的情況下，快取接收到的減少請求的數量。	計數

指標	描述	單位
DecrMisses	在請求的金鑰未找到的情況下，快取接收到的減少請求的數量。	計數
DeleteHits	在請求的金鑰已找到的情況下，快取接收到的刪除請求的數量。	計數
DeleteMisses	在請求的金鑰未找到的情況下，快取接收到的刪除請求的數量。	計數
Evictions	快取已移除以提供空間給新的寫入的未逾期項目的數量。	計數
GetHits	在請求的金鑰已找到的情況下，快取接收到的取得請求的數量。	計數
GetMisses	快取已接收到其請求的金鑰未找到的取得請求的數量。	計數
IncrHits	在請求的金鑰已找到的情況下，快取接收到的增加請求的數量。	計數
IncrMisses	在請求的金鑰未找到的情況下，快取接收到的增加請求的數量。	計數
Reclaimed	快取已移除以提供空間給新的寫入的逾期項目的數量。	計數

針對 Memcached 1.4.14 提供以下額外的指標。

指標	描述	單位
BytesUsedForHash	雜湊表目前使用的位元組數量。	位元組
CmdConfigGet	累積的 config get 請求的數量。	計數
CmdConfigSet	累積的 config set 請求的數量。	計數

指標	描述	單位
CmdTouch	累積的 touch 請求的數量。	計數
CurrConfig	目前已儲存的組態的數量。	計數
EvictedUnfetched	從最近最少使用且在設定後皆未被碰觸的快取 (LRU) 移出的有效項目的數量。	計數
ExpiredUnfetched	從設定後皆未被碰觸的 LRU 回收的逾期項目的數量。	計數
SlabsMoved	已移除的 slab 頁面的總數。	計數
TouchHits	已被碰觸並賦予新的過期時間的金鑰的數量。	計數
TouchMisses	已被碰觸但未被找到的項目的數量。	計數

AWS/命ElastiCache 名空間包含下列計算的快取層級量度。

指標	描述	單位
NewConnections	快取已接收到的新連線的數量。這是衍伸自藉由記錄 total_connections 在一段期間內的變更而獲得的 memcached total_connections 統計資訊。由於保留給 a 的连接，這將始終至少為 1 ElastiCache。	計數
NewItems	快取已儲存的新項目的數量。這是衍伸自藉由記錄 total_items 在一段期間內的變更而獲得的 memcached total_items 統計資訊。	計數
UnusedMemory	未被資料使用的記憶體量。這是衍伸自 Memcached 統計資訊 limit_maxbytes 和藉由減去 limit_maxbytes 位元組而獲得的。  由於 Memcached 的額外負荷會使用資料所使用的記憶體，因此不 UnusedMemory 應視為可用	位元組

指標	描述	單位
	<p>於其他資料的記憶體容量。即使您還有一些未使用的記憶體，也可能會遇到被移出的情形。</p> <p>如需詳細資訊，請參閱 <a href="#">Memcached 項目記憶體使用量</a>。</p>	

## 我應該監控哪些指標？

以下 CloudWatch 指標可提供 ElastiCache 效能的深入分析。在大多數的案例中，我們建議您為這些指標設定 CloudWatch 警示，讓您可以在發生效能問題前先採取修正動作。

### 要監控的指標

- [CPUUtilization](#)
- [SwapUsage](#)
- [移出](#)
- [CurrConnections](#)

### CPUUtilization

此為主機層級指標，以百分比報告。如需更多詳細資訊，請參閱 [主機層級指標](#)。

因為 Memcached 為多執行緒，此指標可高達 90%。若您超過此閾值，請透過使用更大的快取節點類型向上擴展快取叢集，或是透過新增更多快取節點來向外擴展。

### SwapUsage

此為主機層級指標，以位元組報告。如需更多詳細資訊，請參閱 [主機層級指標](#)。

FreeableMemory CloudWatch 指標接近 0 (亦即低於 100MB) 或 SwapUsage 指標大於 FreeableMemory 指標表示節點處於記憶體壓力之下。若是如此，建議您增加 ConnectionOverhead 參數值。

### 移出

此為快取引擎指標。建議您根據應用程式需求，親自判斷此指標的警示閾值。

若您超過所選擇的閾值，請透過使用更大的節點類型向上擴展叢集，或是透過新增更多節點來向外擴展。

### CurrConnections

此為快取引擎指標。建議您根據應用程式需求，親自判斷此指標的警示閾值。

CurrConnections 的數字增加，可能表示您的應用程式發生問題。您需要調查應用程式行為才能處理此問題。

## 監控 CloudWatch 叢集和節點指標

ElastiCache 已與 CloudWatch 整合，因此您可以收集各種指標。您可以使用 CloudWatch 來監控這些指標。

### Note

下列範例需使用 CloudWatch 命令列工具。如需 CloudWatch 和下載開發人員工具的詳細資訊，請參閱 [CloudWatch 產品頁面](#)。

下列程序說明如何使用 CloudWatch 來為快取叢集收集過去一小時的儲存空間統計數字。

### Note

以下範例提供的 StartTime 和 EndTime 值僅做為說明之用。您必須將快取節點的開始和結束時間值替換為適合的值。

如需 ElastiCache 配額的相關資訊，請參閱適用於 ElastiCache 的 [AWS 服務配額](#)。

## 監控 CloudWatch 叢集和節點指標 (主控台)

為快取叢集收集 CPU 使用率統計資料

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 選取您要檢視其指標的快取節點。

### Note

選取 20 個以上的節點時，會停用主控台上的指標檢視。

- a. 在 AWS 管理主控台的 Cache Clusters (快取叢集) 頁面上，按一下一或多個快取叢集的名稱。  
  
快取叢集的詳細資訊頁面隨即出現。
- b. 按一下視窗頂端的 Nodes (節點) 標籤。

- c. 在詳細資訊視窗上的 Nodes (節點) 標籤，選取您要檢視其指標的快取節點。

主控台視窗底部會顯示可用的 CloudWatch 指標清單。

- d. 按一下 CPU Utilization (CPU 使用率) 指標。

隨即開啟 CloudWatch 主控台，並顯示您選取的指標。若要變更顯示的指標，可以使用 Statistic (統計數字) 和 Period (期間) 下拉式清單方塊和 Time Range (時間範圍) 索引標籤。

## 使用 CloudWatch CLI 監控 CloudWatch 叢集和節點指標

為快取叢集收集 CPU 使用率統計資料

- 若為 Linux、macOS 或 Unix：

```
aws cloudwatch get-metric-statistics \  
  --namespace AWS/ElastiCache \  
  --metric-name CPUUtilization \  
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},  
{"Name":"CacheNodeId","Value":"0001"}]' \  
  --statistics=Average \  
  --start-time 2018-07-05T00:00:00 \  
  --end-time 2018-07-06T00:00:00 \  
  --period=3600
```

針對 Windows：

```
aws cloudwatch get-metric-statistics ^  
  --namespace AWS/ElastiCache ^  
  --metric-name CPUUtilization ^  
  --dimensions='[{"Name":"CacheClusterId","Value":"test"},  
{"Name":"CacheNodeId","Value":"0001"}]' ^  
  --statistics=Average ^  
  --start-time 2018-07-05T00:00:00 ^  
  --end-time 2018-07-06T00:00:00 ^  
  --period=3600
```

## 使用 CloudWatch API 監控 CloudWatch 叢集和節點指標

為快取叢集收集 CPU 使用率統計資料

- 使用下列參數呼叫 CloudWatch API GetMetricStatistics (請注意，顯示的開始和結束時間僅為範例；您必須將其替代為適當的開始和結束時間)：
  - 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=mycacheclass,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=mycacheclass"  
  &Dimensions.member.2="CacheNodeId=0002"  
  &Namespace=&AWS;/ElastiCache  
  &MeasureName=CPUUtilization  
  &Timestamp=2018-07-07T17:3A48%3A21.746Z  
  &AWS;AccessKeyId=<&AWS; Access Key ID>  
  &Signature=<Signature>
```

## Amazon SNS 的 ElastiCache 事件監控

當叢集上發生重大事件時，ElastiCache 會向特定的 Amazon SNS 主題傳送通知。範例包含新增節點失敗、新增節點成功、安全群組修改和其他事件。藉由監控重要事件，您可以了解叢集目前的狀態，並根據事件採取正確的動作。

### 主題

- [管理 ElastiCache Amazon SNS 通知](#)
- [檢視 ElastiCache 事件](#)
- [事件通知和 Amazon SNS](#)

## 管理 ElastiCache Amazon SNS 通知

您可以將 ElastiCache 設定為使用 Amazon Simple Notification Service (Amazon SNS) 傳送重要叢集事件的通知。在這些範例中，您會使用 Amazon SNS 主題的 Amazon Resource Name (ARN) 設定叢集以接收通知。

### Note

本主題假設您已註冊 Amazon SNS，並已設定及訂閱 Amazon SNS 主題。如需操作方式的相關資訊，請參閱 [Amazon Simple Notification Service 開發人員指南](#)。

## 新增 Amazon SNS 主題

下列幾節說明如何使用 AWS 主控台、AWS CLI 或 ElastiCache API 來新增 Amazon SNS 主題。

### 新增 Amazon SNS 主題 (主控台)

下列程序示範如何為叢集新增 Amazon SNS 主題。

### Note

此程序也可用於修改 Amazon SNS 主題。

### 為叢集新增或修改 Amazon SNS 主題 (主控台)

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 在 Clusters (叢集) 中，選擇您要新增或修改 Amazon SNS 主題 ARN 的叢集。
3. 選擇 Modify (修改)。
4. 在 Modify Cluster (修改叢集) 的 Topic for SNS Notification (SNS 通知的主題) 下，選擇您要新增的 SNS 主題，或選擇 Manual ARN input (手動輸入 ARN)，並輸入 Amazon SNS 主題的 ARN。
5. 選擇 Modify (修改)。

## 新增 Amazon SNS 主題 (AWS CLI)

若要為叢集新增或修改 Amazon SNS 主題，請使用 AWS CLI 命令 `modify-cache-cluster`。

下列程式碼範例會將 Amazon SNS 主題 ARN 新增至 `my-cluster`。

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-cluster \  
  --cache-cluster-id my-cluster \  
  --notification-topic-arn arn:aws:sns:us-west-2:123456789xxx:ElastiCacheNotifications
```

針對 Windows：

```
aws elasticache modify-cache-cluster ^  
  --cache-cluster-id my-cluster ^  
  --notification-topic-arn arn:aws:sns:us-west-2:123456789xx:ElastiCacheNotifications
```

如需詳細資訊，請參閱 [modify-cache-cluster](#)。

## 新增 Amazon SNS 主題 (ElastiCache API)

若要為叢集新增或修改 Amazon SNS 主題，請搭配下列參數呼叫 `ModifyCacheCluster` 動作：

- `CacheClusterId=my-cluster`
- `TopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications`

## Example

```
https://elasticache.amazon.com/  
  ?Action=ModifyCacheCluster  
  &ApplyImmediately=false  
  &CacheClusterId=my-cluster  
  &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-west-2%3A565419523791%3AElastiCacheNotifications  
  &Version=2014-12-01  
  &SignatureVersion=4  
  &SignatureMethod=HmacSHA256  
  &Timestamp=20141201T220302Z
```

```
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Date=20141201T220302Z
&X-Amz-SignedHeaders=Host
&X-Amz-Expires=20141201T220302Z
&X-Amz-Credential=<credential>
&X-Amz-Signature=<signature>
```

如需詳細資訊，請參閱 [ModifyCacheCluster](#)。

## 啟用和停用 Amazon SNS 通知

您可以為叢集開啟或關閉通知。下列程序示範如何停用 Amazon SNS 通知。

### 啟用和停用 Amazon SNS 通知 (主控台)

使用 AWS Management Console 停用 Amazon SNS 通知

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看執行 Memcached 的叢集清單，請在導覽窗格中選擇 Memcached。
3. 選擇您要修改通知之叢集左邊的方塊。
4. 選擇 Modify (修改)。
5. 在 Modify Cluster (修改叢集) 的 Topic for SNS Notification (SNS 通知的主題) 下，選擇 Disable Notifications (停用通知)。
6. 選擇 Modify (修改)。

### 啟用和停用 Amazon SNS 通知 (AWS CLI)

若要停用 Amazon SNS 通知，請搭配下列參數使用 `modify-cache-cluster` 命令：

若為 Linux、macOS 或 Unix：

```
aws elasticache modify-cache-cluster \
  --cache-cluster-id my-cluster \
  --notification-topic-status inactive
```

針對 Windows：

```
aws elasticache modify-cache-cluster ^
```

```
--cache-cluster-id my-cluster ^  
--notification-topic-status inactive
```

## 啟用和停用 Amazon SNS 通知 (ElastiCache API)

若要停用 Amazon SNS 通知，請搭配下列參數呼叫 `ModifyCacheCluster` 動作：

- `CacheClusterId=my-cluster`
- `NotificationTopicStatus=inactive`

此呼叫會傳回類似以下的輸出：

### Example

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

## 檢視 ElastiCache 事件

ElastiCache 會記錄與您叢集執行個體、安全群組及參數群組相關的事件。此資訊包含事件的日期和時間、事件的來源名稱和來源類型，以及事件的描述。您可以使用 ElastiCache 主控台、AWS CLI `describe-events` 命令或 ElastiCache API 動作 `DescribeEvents`，輕鬆從日誌擷取事件。

下列程序示範如何檢視過去 24 小時 (1440 分鐘) 內的所有 ElastiCache 事件。

### 檢視 ElastiCache 事件 (主控台)

下列程序使用 ElastiCache 主控台顯示事件。

#### 使用 ElastiCache 主控台檢視事件

1. 登入 AWS Management Console 並前往 <https://console.aws.amazon.com/elasticache/> 開啟 ElastiCache 主控台。
2. 若要查看所有可用事件的清單，請在導覽窗格中選擇 Events (事件)。

在 Events (事件) 畫面上，清單的每一列各代表一個事件，並會顯示事件來源、事件類型 (`cache-cluster`、`cache-parameter-group`、`cache-security-group` 或 `cache-subnet-group`)、事件的 GMT 時間，以及事件的描述。

您可以使用 Filter (篩選條件) 指定要查看事件清單中的所有事件，還是只查看特定類型的事件。

### 檢視 ElastiCache 事件 (AWS CLI)

若要使用 AWS CLI 產生 ElastiCache 事件的清單，請使用 `describe-events` 命令。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼最多會列出 40 個快取叢集事件。

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內的所有事件。

```
aws elasticache describe-events --source-type cache-cluster --duration 1440
```

`describe-events` 命令的輸出會類似下列內容。

```
aws elasticache describe-events --source-type cache-cluster --max-items 40
{
```

```
"Events": [  
  {  
    "SourceIdentifier": "my-mem-cluster",  
    "SourceType": "cache-cluster",  
    "Message": "Finished modifying number of nodes from 1 to 3",  
    "Date": "2020-06-09T02:01:21.772Z"  
  },  
  {  
    "SourceIdentifier": "my-mem-cluster",  
    "SourceType": "cache-cluster",  
    "Message": "Added cache node 0002 in availability zone us-west-2a",  
    "Date": "2020-06-09T02:01:21.716Z"  
  },  
  {  
    "SourceIdentifier": "my-mem-cluster",  
    "SourceType": "cache-cluster",  
    "Message": "Added cache node 0003 in availability zone us-west-2a",  
    "Date": "2020-06-09T02:01:21.706Z"  
  },  
  {  
    "SourceIdentifier": "my-mem-cluster",  
    "SourceType": "cache-cluster",  
    "Message": "Increasing number of requested nodes",  
    "Date": "2020-06-09T01:58:34.178Z"  
  },  
  {  
    "SourceIdentifier": "mycluster-0003-004",  
    "SourceType": "cache-cluster",  
    "Message": "Added cache node 0001 in availability zone us-west-2c",  
    "Date": "2020-06-09T01:51:14.120Z"  
  },  
  {  
    "SourceIdentifier": "mycluster-0003-004",  
    "SourceType": "cache-cluster",  
    "Message": "This cache cluster does not support persistence (ex:  
'appendonly'). Please use a different instance type to enable persistence.",  
    "Date": "2020-06-09T01:51:14.095Z"  
  },  
  {  
    "SourceIdentifier": "mycluster-0003-004",  
    "SourceType": "cache-cluster",  
    "Message": "Cache cluster created",  
    "Date": "2020-06-09T01:51:14.094Z"  
  },  
]
```

```
{
  "SourceIdentifier": "mycluster-0001-005",
  "SourceType": "cache-cluster",
  "Message": "Added cache node 0001 in availability zone us-west-2b",
  "Date": "2020-06-09T01:42:55.603Z"
},
{
  "SourceIdentifier": "mycluster-0001-005",
  "SourceType": "cache-cluster",
  "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
  "Date": "2020-06-09T01:42:55.576Z"
},
{
  "SourceIdentifier": "mycluster-0001-005",
  "SourceType": "cache-cluster",
  "Message": "Cache cluster created",
  "Date": "2020-06-09T01:42:55.574Z"
},
{
  "SourceIdentifier": "mycluster-0001-004",
  "SourceType": "cache-cluster",
  "Message": "Added cache node 0001 in availability zone us-west-2b",
  "Date": "2020-06-09T01:28:40.798Z"
},
{
  "SourceIdentifier": "mycluster-0001-004",
  "SourceType": "cache-cluster",
  "Message": "This cache cluster does not support persistence (ex:
'appendonly'). Please use a different instance type to enable persistence.",
  "Date": "2020-06-09T01:28:40.775Z"
},
{
  "SourceIdentifier": "mycluster-0001-004",
  "SourceType": "cache-cluster",
  "Message": "Cache cluster created",
  "Date": "2020-06-09T01:28:40.773Z"
}
]
```

如需可用參數和允許參數值這類項目的詳細資訊，請參閱 [describe-events](#)。

## 檢視 ElastiCache 事件 (ElastiCache API)

若要使用 ElastiCache API 產生 ElastiCache 事件的清單，請使用 DescribeEvents 動作。您可以使用選用參數來控制列出的事件類型、列出的事件時間範圍，要列出的最大事件數等等。

下列程式碼會列出 40 個最新的快取叢集事件。

```
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>
```

下列程式碼會列出過去 24 小時 (1440 分鐘) 內的快取叢集事件。

```
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>
```

上述動作產生的輸出應該會類似下列內容。

```
<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>
```

如需可用參數和允許參數值這類項目的詳細資訊，請參閱 [DescribeEvents](#)。

## 事件通知和 Amazon SNS

快取叢集上發生重要事件時，ElastiCache 可以使用 Amazon Simple Notification Service (SNS) 發佈訊息。此功能可用於重新整理用戶端機器上的伺服器清單，這些機器連線至快取叢集的個別快取節點端點。

### Note

如需 Amazon Simple Notification Service (SNS) 的詳細資訊，包含定價資訊和 Amazon SNS 說明文件的連結，請參閱 [Amazon SNS 產品頁面](#)。

通知會發佈至特定的 Amazon SNS 主題。下列是通知的需求：

- ElastiCache 通知只能設定一個主題。
- 擁有該 Amazon SNS 主題的 AWS 帳戶，必須與擁有已啟用通知之快取叢集的帳戶相同。
- 您要發佈的 Amazon SNS 主題無法加密。

### Note

加密 (靜態) 的 Amazon SNS 主題可以連接到叢集。不過，ElastiCache 主控台的主題狀態會顯示為非作用中，這樣可在 ElastiCache 將訊息推送至主題時有效取消主題與叢集的關聯。

- Amazon SNS 主題所在區域必須與 ElastiCache 叢集所在區域相同。

## ElastiCache 事件

下列 ElastiCache 事件會觸發 Amazon SNS 通知。如需事件的詳細資訊，請參閱 [檢視 ElastiCache 事件](#)。

事件名稱	訊息	描述
ElastiCache:AddCacheNodeComplete	ElastiCache:AddCacheNodeComplete : <i>cache-cluster</i>	快取節點已新增至快取叢集，且已就緒可供使用。
ElastiCache:AddCacheNodeFailed，因可用的 IP 地址不足所導致	ElastiCache:AddCacheNodeFailed : <i>cluster-name</i>	由於可用的 IP 地址不足，而無法新增快取節點。
ElastiCache:CacheClusterParametersChanged	ElastiCache:CacheClusterParametersChanged : <i>cluster-name</i>	一或多個快取叢集參數已變更。
ElastiCache:CacheClusterProvisioningComplete	ElastiCache:CacheClusterProvisioningComplete <i>cluster-name-0001-005</i>	快取叢集的佈建已完成，且快取叢集中的快取節點已就緒並可供使用。
ElastiCache:CacheClusterProvisioningFailed，因不相容的網路狀態所導致	ElastiCache:CacheClusterProvisioningFailed : <i>cluster-name</i>	已嘗試將新快取叢集啟動至不存在的虛擬私有雲端 (VPC)。
ElastiCache:CacheClusterScalingComplete	CacheClusterScalingComplete : <i>cluster-name</i>	快取叢集的擴展已成功完成。
ElastiCache:CacheClusterScalingFailed	ElastiCache:CacheClusterScalingFailed: <i>cluster-name</i>	快取叢集上的擴展操作失敗。
ElastiCache:CacheClusterSecurityGroupModified	ElastiCache:CacheClusterSecurityGroupModified : <i>cluster-name</i>	已發生下列其中一項事件： <ul style="list-style-type: none"> <li>• 授權給快取叢集的快取安全群組清單已修改。</li> <li>•</li> </ul>

事件名稱	訊息	描述
		<p>一或多個新 EC2 安全群組，已在與快取叢集相關聯的任何快取安全群組上授權。</p> <ul style="list-style-type: none"> <li>一或多個新 EC2 安全群組，已從與快取叢集相關聯的任何快取安全群組上撤銷。</li> </ul>
ElastiCache:CacheNodeReplaceStarted	ElastiCache:CacheNodeReplaceStarted : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機已降級或無法連線，且已開始取代快取節點。</p> <div data-bbox="1068 772 1507 995" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>取代之快取節點的 DNS 項目並未變更。</p> </div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。然而，某些快取用戶端程式庫可能會在 ElastiCache 取代快取節點後停止使用快取節點；在這種情況下，應用程式應在此事件發生時重新整理伺服器清單。</p>

事件名稱	訊息	描述
ElastiCache:CacheNodeReplaceComplete	ElastiCache:CacheNodeReplaceComplete : <i>cluster-name</i>	<p>ElastiCache 偵測到執行快取節點的主機已降級或無法連線，且已完成取代快取節點。</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>取代之快取節點的 DNS 項目並未變更。</p> </div> <p>在多數情況下，當發生此事件時，您不需要重新整理用戶端的伺服器清單。然而，某些快取用戶端程式庫可能會在 ElastiCache 取代快取節點後停止使用快取節點；在這種情況下，應用程式應在此事件發生時重新整理伺服器清單。</p>
ElastiCache:CacheNodesRebooted	ElastiCache:CacheNodesRebooted : <i>cluster-name</i>	<p>一或多個快取節點已重新開機。</p> <p>訊息 (Memcached) : "Cache node %s shutdown" 然後，第二個訊息 : "Cache node %s restarted"</p>
ElastiCache:CertificateRenewalComplete (僅 Redis)	ElastiCache:CertificateRenewalComplete	Amazon CA 憑證已成功續約。
ElastiCache:CreateReplicationGroupComplete	ElastiCache:CreateReplicationGroupComplete : <i>cluster-name</i>	已成功建立複寫群組。

事件名稱	訊息	描述
ElastiCache:DeleteCacheClusterComplete	ElastiCache:DeleteCacheClusterComplete : <i>cluster-name</i>	已完成刪除快取叢集和所有與其相關聯的快取節點。
ElastiCache:FailoverComplete (僅 Redis)	ElastiCache:FailoverComplete : <i>mycluster</i>	已成功容錯移轉至複本節點。
ElastiCache:ReplicationGroupIncreaseReplicaCountFinished	ElastiCache:ReplicationGroupIncreaseReplicaCountFinished : <i>cluster-name-0001-005</i>	叢集中的複本數量已增加。
ElastiCache:ReplicationGroupIncreaseReplicaCountStarted	ElastiCache:ReplicationGroupIncreaseReplicaCountStarted : <i>cluster-name-0003-004</i>	將複本新增至叢集的程序已經開始。
ElastiCache:NodeReplacementCanceled	ElastiCache:NodeReplacementCanceled : <i>cluster-name</i>	叢集中原先已排程替換的節點，已不再排程替換。
ElastiCache:NodeReplacementRescheduled	ElastiCache:NodeReplacementRescheduled : <i>cluster-name</i>	<p>叢集中原先已排程替換的節點，已重新排程在通知中所述的新視窗期間進行替換。</p> <p>如需您可以採取哪些動作的相關資訊，請參閱<a href="#">取代 Memcached 的快取節點</a>。</p>

事件名稱	訊息	描述
ElastiCache:NodeReplacementScheduled	ElastiCache:NodeReplacementScheduled : <i>cluster-name</i>	您叢集中的節點，已排在通知中所述的視窗期間進行替換。  如需您可以採取哪些動作的相關資訊，請參閱 <a href="#">取代 Memcached 的快取節點</a> 。
ElastiCache:RemoveCacheNodeComplete	ElastiCache:RemoveCacheNodeComplete : <i>cluster-name</i>	快取節點已從快取叢集移除。
ElastiCache:ReplicationGroupScalingComplete	ElastiCache:ReplicationGroupScalingComplete : <i>cluster-name</i>	已順利完成複寫群組上的向上擴展操作。
ElastiCache:ReplicationGroupScalingFailed	"Failed applying modification to cache node type to %s."	複寫群組上的擴展操作失敗。
ElastiCache:ServiceUpdateAvailableForNode	"Service update is available for cache node %s."	節點有可用的自助更新。
ElastiCache:SnapshotComplete (僅 Redis)	ElastiCache:SnapshotComplete : <i>cluster-name</i>	快取快照已順利完成。
ElastiCache:SnapshotFailed (僅 Redis)	SnapshotFailed : <i>cluster-name</i>	快取快照已失敗。請參閱叢集的快取事件，以取得失敗原因的詳細資訊。  如果您描述快照，請查看 <a href="#">DescribeSnapshots</a> ，其狀態將是 failed。

## 相關主題

- [檢視 ElastiCache 事件](#)

# 使用 AWS CloudTrail 記錄 Amazon ElastiCache API 呼叫

Amazon ElastiCache 已與 AWS CloudTrail 整合，這項服務可提供由使用者、角色或 Amazon ElastiCache 中 AWS 服務所採取之動作的記錄。CloudTrail 將 Amazon ElastiCache 的所有 API 呼叫擷取為事件，包括來自 Amazon ElastiCache 主控台的呼叫以及來自對 Amazon ElastiCache API 作業發出的程式碼呼叫。如果您建立追蹤記錄，就可以持續傳送 CloudTrail 事件至 Amazon S3 儲存貯體，包括 Amazon ElastiCache 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台當中的 Event history (事件歷史記錄) 檢視最新事件。您可以利用 CloudTrail 所收集的資訊來判斷向 Amazon ElastiCache 發出的請求，以及發出請求的 IP 地址、人員、時間和其他詳細資訊。

若要進一步了解 CloudTrail，請參閱 [《AWS CloudTrail 使用者指南》](#)。

## CloudTrail 中的 Amazon ElastiCache 資訊

當您建立帳戶時，系統即會在 AWS 帳戶中啟用 CloudTrail。在 Amazon ElastiCache 中發生活動時，該活動將與 Event history (事件歷史紀錄) 中的其他 AWS 服務事件一起記錄在 CloudTrail 事件中。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱 [使用 CloudTrail 事件歷史記錄檢視事件](#)。

若要持續記錄您 AWS 帳戶中正在進行的事件 (包括 Amazon ElastiCache 的事件)，請建立追蹤記錄。追蹤能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台建立權杖時，權杖會套用到所有區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案](#)

CloudTrail 會記錄所有 Amazon ElastiCache 動作，並記載於 [ElastiCache API 參考](#) 中。例如，對 CreateCacheCluster、DescribeCacheCluster 和 ModifyCacheCluster 動作發出的呼叫會在 CloudTrail 記錄檔案中產生項目。

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

## 了解 Amazon ElastiCache 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔案並非依公有 API 呼叫追蹤記錄的堆疊排序，因此不會以任何特定順序出現。

以下範例顯示的是展示 CreateCacheCluster 動作的 CloudTrail 日誌項目。

```
{
  "eventVersion": "1.01",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLEEXAMPLEEXAMPLE",
    "arn": "arn:aws:iam:123456789012:user/elasticache-allow",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "elasticache-allow"
  },
  "eventTime": "2014-12-01T22:00:35Z",
  "eventSource": "elasticache.amazonaws.com",
  "eventName": "CreateCacheCluster",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.01",
  "userAgent": "AWS CLI/ElastiCache 1.10 API 2014-12-01",
  "requestParameters": {
    "numCacheNodes": 2,
    "cacheClusterId": "test-memcached",
    "engine": "memcached",
    "aZMode": "cross-az",
    "cacheNodeType": "cache.m1.small",
  },
}
```

```

    "responseElements":{
      "engine":"memcached",
      "clientDownloadLandingPage":"https://console.aws.amazon.com/elasticache/
home#client-download:",
      "cacheParameterGroup":{
        "cacheParameterGroupName":"default.memcached1.4",
        "cacheNodeIdsToReboot":{
          },
        "parameterApplyStatus":"in-sync"
      },
      "preferredAvailabilityZone":"Multiple",
      "numCacheNodes":2,
      "cacheNodeType":"cache.m1.small",

      "cacheClusterStatus":"creating",
      "autoMinorVersionUpgrade":true,
      "preferredMaintenanceWindow":"thu:05:00-thu:06:00",
      "cacheClusterId":"test-memcached",
      "engineVersion":"1.4.14",
      "cacheSecurityGroups":[
        {
          "status":"active",
          "cacheSecurityGroupName":"default"
        }
      ],
      "pendingModifiedValues":{
        }
    },
    "requestID":"104f30b3-3548-11e4-b7b8-6d79ffe84edd",
    "eventID":"92762127-7a68-42ce-8787-927d2174cde1"
  }

```

以下範例顯示的是展示 DescribeCacheCluster 動作的 CloudTrail 日誌項目。請注意，所有 Amazon ElastiCache Describe 呼叫 (ResponseElements) 都已移除 Describe\* 區段，並顯示為 null。

```

{
  "eventVersion":"1.01",
  "userIdentity":{
    "type":"IAMUser",
    "principalId":"EXAMPLEEXAMPLEEXAMPLE",
    "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId":"123456789012",

```

```

    "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
    "userName":"elasticache-allow"
  },
  "eventTime":"2014-12-01T22:01:00Z",
  "eventSource":"elasticache.amazonaws.com",
  "eventName":"DescribeCacheClusters",
  "awsRegion":"us-west-2",
  "sourceIPAddress":"192.0.2.01",
  "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
  "requestParameters":{
    "showCacheNodeInfo":false,
    "maxRecords":100
  },
  "responseElements":null,
  "requestID":"1f0b5031-3548-11e4-9376-c1d979ba565a",
  "eventID":"a58572a8-e81b-4100-8e00-1797ed19d172"
}

```

以下範例顯示的是記錄 `ModifyCacheCluster` 動作的 CloudTrail 日誌項目。

```

{
  "eventVersion":"1.01",
  "userIdentity":{
    "type":"IAMUser",
    "principalId":"EXAMPLEEXAMPLEEXAMPLE",
    "arn":"arn:aws:iam::123456789012:user/elasticache-allow",
    "accountId":"123456789012",
    "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
    "userName":"elasticache-allow"
  },
  "eventTime":"2014-12-01T22:32:21Z",
  "eventSource":"elasticache.amazonaws.com",
  "eventName":"ModifyCacheCluster",
  "awsRegion":"us-west-2",
  "sourceIPAddress":"192.0.2.01",
  "userAgent":"AWS CLI/ElastiCache 1.10 API 2014-12-01",
  "requestParameters":{
    "applyImmediately":true,
    "numCacheNodes":3,
    "cacheClusterId":"test-memcached"
  },
  "responseElements":{
    "engine":"memcached",

```

```
    "clientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/home#client-download:",
    "cacheParameterGroup": {
      "cacheParameterGroupName": "default.memcached1.4",
      "cacheNodeIdsToReboot": {
      },
      "parameterApplyStatus": "in-sync"
    },
    "cacheClusterCreateTime": "Dec 1, 2014 10:16:06 PM",
    "preferredAvailabilityZone": "Multiple",
    "numCacheNodes": 2,
    "cacheNodeType": "cache.m1.small",
    "cacheClusterStatus": "modifying",
    "autoMinorVersionUpgrade": true,
    "preferredMaintenanceWindow": "thu:05:00-thu:06:00",
    "cacheClusterId": "test-memcached",
    "engineVersion": "1.4.14",
    "cacheSecurityGroups": [
      {
        "status": "active",
        "cacheSecurityGroupName": "default"
      }
    ],
    "configurationEndpoint": {
      "address": "test-memcached.example.cfg.use1prod.cache.amazonaws.com",
      "port": 11211
    },
    "pendingModifiedValues": {
      "numCacheNodes": 3
    }
  },
  "requestID": "807f4bc3-354c-11e4-9376-c1d979ba565a",
  "eventID": "e9163565-376f-4223-96e9-9f50528da645"
}
```

## ElastiCache 的配額

對於每個 AWS 服務，您的 AWS 帳戶有預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定規定。您可以請求提高某些配額，而其他配額無法提高。

若要檢視 ElastiCache 的配額，請開啟 [Service Quotas 主控台](#)。在導覽窗格中，選擇 AWS services (AWS 服務)，然後選取 ElastiCache。

若要請求增加配額，請參閱 Service Quotas 使用者指南中的 [請求提高配額](#)。如果 Service Quotas 中尚未提供配額，請使用 [增加服務配額表單](#)。

您的 AWS 帳戶具有下列與 ElastiCache 相關的配額。

資源	預設
每個區域的無伺服器快取	40
每個區域的節點	300
每個叢集的節點數	40
每區域參數群組數	300
每區域安全群組數	50
每區域子網路群組數	300
每個子網路群組的子網路	20

## 參考

本節中的主題涵蓋使用 Amazon ElastiCache API 的內容，以及 ElastiCache 的 AWS CLI 章節。本節也包含常見錯誤訊息及服務通知。

- [使用 ElastiCache API](#)
- [ElastiCache API 參考](#)
- [AWS CLI 參考的 ElastiCache 章節](#)
- [Amazon ElastiCache 錯誤訊息](#)
- [通知](#)

## 使用 ElastiCache API

本節提供如何使用和實作 ElastiCache 操作的任務導向說明。如需關於這些作業的完整說明，請參閱 [Amazon ElastiCache API 參考](#)

### 主題

- [使用查詢 API](#)
- [可用程式庫](#)
- [對應用程式進行疑難排解](#)

## 使用查詢 API

### 查詢參數

HTTP 查詢型請求是使用 HTTP 動詞 GET 或 POST，以及名為 Action 之查詢參數的 HTTP 請求。

每一個查詢請求都須包括一些常用參數，來處理動作的身分驗證和選取。

部分操作有數個參數清單。這些清單是使用 param.*n* 表示法來指定的。*n* 的值是從 1 開始的整數。

### 查詢請求身分驗證

您只能透過 HTTPS 傳送查詢請求，而且必須在每一個查詢請求中包括一個簽章。本節說明如何建立簽章。下列程序所述的方法也稱為「簽章第 4 版」。

下列為用來對 AWS 進行驗證要求的基本步驟。此處假設您已向 AWS 註冊，並具有存取金鑰 ID 與私密存取金鑰。

### 查詢身分驗證程序

1. 寄件者可編寫傳給 AWS 的請求。
2. 寄件者會計算請求簽章，其為使用 SHA-1 雜湊函數的雜湊型訊息身分驗證程式碼 (HMAC) 的金鑰式雜湊，如本主題下一節所述。
3. 請求寄件者會將請求資料、簽章和存取金鑰 ID (所用之私密存取金鑰的金鑰識別碼) 傳送給 AWS。
4. AWS 會使用存取金鑰 ID 來查詢私密存取金鑰。
5. AWS 使用的演算法與計算請求中簽章的演算法相同，可從請求資料及私密存取金鑰產生簽章。
6. 如果簽章相符，則會將請求視為真實請求。若比較失敗，則會捨棄該要求，且 AWS 會傳回錯誤回應。

#### Note

如果請求包含 `Timestamp` 參數，針對請求計算出的簽章就會在其值的 15 分鐘後過期。  
如果請求包含 `Expires` 參數，簽章就會於 `Expires` 參數指定的時間過期。

### 計算請求簽章

1. 建立標準化查詢字串，以在本程序稍後使用：
  - a. 依據參數名稱與自然位元組順序，排序 UTF-8 查詢字串元件。參數可以來自 GET URI 或 POST 內文 (當 `Content-Type` 為 `application/x-www-form-urlencoded` 時)。
  - b. 根據下列規則，為每個參數名稱和值執行 URL 編碼：
    - i. 請不要針對 RFC 3986 所定義的任何未預留字元執行 URL 編碼。這些未預留字元包括 A-Z、a-z、0-9、連字號 (-)、底線 (\_)、句點 (.) 及波狀符號 (~)。
    - ii. 對所有其他含有 `%XY` 的字元執行百分比編碼，其中 X 和 Y 是十六進位字元 0-9 和大寫 A-F。
    - iii. 對 `%XY%ZA...` 格式的延伸 UTF-8 字元執行百分比編碼。
    - iv. 將空白字元百分比編碼為 `%20` (而非常見編碼機制使用的 `+`)。

- c. 使用等號 (=) (ASCII 字元 61) 分隔編碼過的參數名稱與其編碼值，即使該參數值為空白也是如此。
  - d. 使用 & 符號 (ASCII 字碼 38) 分隔名稱值對。
2. 依據下列虛擬語法 ("\n" 代表 ASCII 換行符號)，建立要簽署的字串。

```
StringToSign = HTTPVerb + "\n" +  
ValueOfHostHeaderInLowercase + "\n" +  
HTTPRequestURI + "\n" +  
CanonicalizedQueryString <from the preceding step>
```

HTTPRequestURI 元件是 URI 的 HTTP 絕對路徑元件，一直到查詢字串為止 (但不包括查詢字串)。如果 HTTPRequestURI 空白，請使用斜線 (/)。

3. 使用您剛建立的字串、私密存取金鑰做為金鑰，以及 SHA256 或 SHA1 做為雜湊演算法，計算 RFC 2104 相容的 HMAC。

如需詳細資訊，請參閱 <https://www.ietf.org/rfc/rfc2104.txt>。

4. 將結果值轉換成 base64。
5. 將該值包含為請求中的 Signature 參數值。

例如，以下是範例請求 (為了清楚起見，已加入分行符號)。

```
https://elasticache.us-west-2.amazonaws.com/  
?Action=DescribeCacheClusters  
&CacheClusterIdentifier=myCacheCluster  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4  
&Version=2014-12-01
```

針對上述的查詢字串，您可透過下列字串來計算 HMAC 簽章。

```
GET\n  
elasticache.amazonaws.com\n  
Action=DescribeCacheClusters  
&CacheClusterIdentifier=myCacheCluster  
&SignatureMethod=HmacSHA256  
&SignatureVersion=4
```

```
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE%2F20140523%2Fus-west-2%2Felasticache
%2Faws4_request
&X-Amz-Date=20141201T223649Z
&X-Amz-SignedHeaders=content-type%3Bhost%3Buser-agent%3Bx-amz-content-sha256%3Bx-
amz-date
    content-type:
    host:elasticache.us-west-2.amazonaws.com
    user-agent:CacheServicesAPICommand_Client
x-amz-content-sha256:
x-amz-date:
```

結果為下列簽署的請求。

```
https://elasticache.us-west-2.amazonaws.com/
?Action=DescribeCacheClusters
&CacheClusterIdentifier=myCacheCluster
&SignatureMethod=HmacSHA256
&SignatureVersion=4
&Version=2014-12-01
&X-Amz-Algorithm=&AWS;4-HMAC-SHA256
&X-Amz-Credential=AKIADQKE4SARGYLE/20141201/us-west-2/elasticache/aws4_request
&X-Amz-Date=20141201T223649Z
&X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
&X-Amz-Signature=2877960fced9040b41b4feaca835fd5cfeb9264f768e6a0236c9143f915ffa56
```

如需簽章程序和計算請求簽章的詳細資訊，請參閱 [Signature 第 4 版簽署程序](#) 主題和副主題。

## 可用程式庫

針對偏好使用特定語言 API (而非查詢 API) 建置應用程式的開發人員，AWS 提供軟體開發套件 (SDK)。這些軟體開發套件提供 API 中不包含的基本功能，例如請求身分驗證、請求重試與錯誤處理，以便輕鬆上手。我們提供以下程式設計語言的軟體開發套件和其他資源：

- [Java](#)
- [Windows 與 .NET](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)

如需其他語言的資訊，請參閱[範本程式碼與程式庫](#)。

## 對應用程式進行疑難排解

ElastiCache 會提供特定和描述性錯誤，以協助您在與 ElastiCache API 互動時進行疑難排解。

### 擷取錯誤

通常，您想要應用程式在您花費任何時間處理結果之前，先檢查請求是否已產生錯誤。若要了解系統是否發生錯誤，最簡單的方式即為在 ElastiCache API 的回應中，尋找 Error 節點。

XPath 語法提供簡單的方式，讓您可搜尋 Error 節點是否存在，並輕鬆擷取錯誤碼和訊息。下列程式碼片段使用 Perl 和 XML::XPath 模組，來判斷請求期間是否發生錯誤。如果發生錯誤，程式碼會列印回應中的第一個錯誤碼和訊息。

```
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"; }
```

### 對秘訣進行故障診斷

建議透過下列程序來診斷並解決 ElastiCache API 發生的問題。

- 確認 ElastiCache 正常運作。

若要執行此操作，您只要開啟瀏覽器視窗，並提交查詢請求給 ElastiCache 服務 (例如 <https://elasticache.amazon.com>) 即可。若出現 `MissingAuthenticationTokenException` 或 `500 Internal Server Error`，即確認服務可供使用，並可回應請求。

- 檢查請求的結構。

在 ElastiCache API 參考中，每項 ElastiCache 作業都有一個參考頁面。再次檢查您是否正確使用參數。若要提供有關可能出錯的概念，請查看範例請求或使用者案例，來查看那些範例是否執行類似操作。

- 查看論壇。

ElastiCache 具有開發論壇，您可以在其中搜尋其他人在過程中所遇到問題的解決方案。若要檢視論壇，請參閱

<https://forums.aws.amazon.com/>。

## 設定 ElastiCache 命令列界面

本節說明執行命令列工具的事前準備、如何取得命令列工具、如何設定工具及其環境，以及使用工具的一系列常見範例。

只在您要使用 ElastiCache 的 AWS CLI 時，才需依照本主題中的說明操作。

### Important

在 API 2014-09-30 版之後，Amazon ElastiCache 命令列界面 (CLI) 不支援任何 ElastiCache 改善項目。若要從命令列使用 ElastiCache 功能，請使用 [AWS 命令列介面](#)。

### 主題

- [先決條件](#)
- [取得命令列工具](#)
- [設定工具](#)
- [為工具提供憑證](#)
- [環境變數](#)

## 先決條件

本文件假設您作業於 Linux/UNIX 或 Windows 環境。Amazon ElastiCache 命令列工具亦可正常運作於 Mac OS X 系統 (一種 UNIX 環境)，但本指南未包含具體的 Mac OS X 說明。

以慣例而言，所有命令列文字前方都會加上通用的 **PROMPT>** 命令列提示字元，您機器實際的命令列提示字元可能有所不同。我們也會使用 **\$** 以表示 Linux/UNIX 的特定命令，**C:\>** 則表示 Windows 的特定命令。此命令的範例輸出則直接列出，前方沒有任何前綴字。

## Java 執行階段環境

本指南使用的命令列工具，需要 Java (版本 5 或更新版本) 來執行。您可以安裝 JRE 或 JDK。若要檢視並下載適用於各種平台 (包括 Linux/UNIX 和 Windows) 的 JRE，請參閱 [Java SE Downloads](#)。

## 設定 Java Home 變數

命令列工具依存環境變數 (JAVA\_HOME) 來尋找 Java Runtime。此環境變數應設為包含名為 bin 子目錄的目錄完整路徑，該子目錄包含可執行的 java (Linux 或 UNIX) 或 java.exe (Windows) 可執行檔。

### 設定 Java Home 變數

#### 1. 設定 Java Home 變數。

- 若使用的是 Linux 和 UNIX，請輸入下列命令：

```
$ export JAVA_HOME=<PATH>
```

- 若使用的是 Windows，請輸入下列命令：

```
C:\> set JAVA_HOME=<PATH>
```

#### 2. 執行 `$JAVA_HOME/bin/java -version` 並檢查輸出，來確認路徑設定。

- 在 Linux/UNIX 上，您將會看到類似以下的輸出：

```
$ $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)
```

- 在 Windows 上，您將會看到類似以下的輸出：

```
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)
```

## 取得命令列工具

命令列工具可透過位於 [ElastiCache 開發人員工具網站](#) 上的 ZIP 檔取得。這些工具都是以 Java 編寫，並且包含 Windows 2000/XP/Vista/Windows 7、Linux/UNIX 及 Mac OSX 的殼層指令碼。ZIP 檔案是獨立式的，因此不需要任何安裝。您只需下載 zip 檔案，並將它解壓縮到位於您本機電腦上的目錄。

## 設定工具

命令列工具仰賴環境變數 (AWS\_ELASTICACHE\_HOME) 來尋找支援程式庫。在您使用此工具之前，您必須先設定此環境變數。將它設為您解壓縮命令列工具的目錄路徑。此目錄名為 ElastiCacheCli-A.B.nnnn (A、B 及 n 是版本/發行編號)，並且包含名為 bin 及 lib 的子目錄。

### 設定 AWS\_ELASTICACHE\_HOME 環境變數

- 開啟命令列視窗，然後輸入以下其中一個命令來設定 AWS\_ELASTICACHE\_HOME 環境變數。
  - 若使用的是 Linux 和 UNIX，請輸入下列命令：

```
$ export &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

- 若使用的是 Windows，請輸入下列命令：

```
C:\> set &AWS;_ELASTICACHE_HOME=<path-to-tools>
```

為了使工具更易於使用，我們建議您將工具的 BIN 目錄新增到您的系統 PATH。本指南的其餘部分都假設 BIN 目錄位於您的系統路徑中。

### 將工具的 BIN 目錄新增到您的系統路徑

- 輸入以下命令來將工具的 BIN 目錄新增到您的系統 PATH。
  - 若使用的是 Linux 和 UNIX，請輸入下列命令：

```
$ export PATH=$PATH:$&AWS;_ELASTICACHE_HOME/bin
```

- 若使用的是 Windows，請輸入下列命令：

```
C:\> set PATH=%PATH%;%&AWS;_ELASTICACHE_HOME%\bin
```

**Note**

Windows 環境變數會在您關閉命令視窗時重設。建議您將它設為永久值。請查閱您 Windows 版本的文件以取得詳細資訊。

**Note**

包含空格的路徑必須包裝在雙引號中，例如：  
"C:\Program Files\Java"

## 為工具提供憑證

命令列工具需要您 AWS 帳戶隨附的 AWS 存取金鑰和私密存取金鑰。您可以透過使用命令列或從位於您本機系統上的登入資料檔案來取得它們。

部署包含一個範本檔案 `#{AWS_ELASTICACHE_HOME}/credential-file-path.template`，您需要使用您的資訊編輯此檔案。以下是範本檔案的內容：

```
AWSAccessKeyId=<Write your AWS access ID>  
AWSSecretKey=<Write your AWS secret key>
```

**⚠ Important**

在 UNIX 上，請限制登入資料檔案擁有者的許可權限：

```
$ chmod 600 <the file created above>
```

使用憑證檔案進行設定時，您需要設定 `AWS_CREDENTIAL_FILE` 環境變數，如此 ElastiCache 工具即可找到您的資訊。

### 設定 `AWS_CREDENTIAL_FILE` 環境變數

#### 1. 設定 環境變數：

- 在 Linux 和 UNIX 上，請使用下列命令來更新變數：

```
$ export &AWS;_CREDENTIAL_FILE=<the file created above>
```

- 在 Windows 上，請使用下列命令來設定變數：

```
C:\> set &AWS;_CREDENTIAL_FILE=<the file created above>
```

2. 檢查您的設定正常運作，請執行以下命令：

```
elasticache --help
```

您應該會看到所有 ElastiCache 命令的用量頁面。

## 環境變數

環境變數在編寫指令碼、設定預設值或暫時覆寫預設值時很有用。

除了 `AWS_CREDENTIAL_FILE` 環境變數之外，大多數 ElastiCache 命令列界面隨附的 API 工具都支援以下變數：

- `EC2_REGION` - 要使用的 AWS 區域。
- `AWS_ELASTICACHE_URL` - 要用於服務呼叫的 URL。若已指定 `EC2_REGION` 或傳遞 `--region` 參數，則不需要指定不同區域端點。

以下範例會示範如何設定環境變數 `EC2_REGION` 來設定 API 工具所使用的區域：

Linux、OS X，或 Unix

```
$ export EC2_REGION=us-west-1
```

Windows

```
$ set EC2_REGION=us-west-1
```

## Amazon ElastiCache 錯誤訊息

Amazon ElastiCache 會傳回下列錯誤訊息。您可能會收到 ElastiCache、其他 AWS 服務或 Memcached 傳回的其他錯誤訊息。如需來自 ElastiCache 以外來源的錯誤訊息說明，請參閱產生該錯誤訊息的來源文件。

- [Cluster node quota exceeded](#)
- [Customer's node quota exceeded](#)
- [Insufficient cache cluster capacity](#)

錯誤訊息：超過叢集節點配額。此區域中每個叢集最多可以有 %n 個節點。

原因：您嘗試建立或修改某個叢集，結果該叢集含有的節點超過 %n 個。

解決方案：變更要求，讓叢集的節點數不超過 %n 個。或者，如果您需要超過 %n 個節點，請使用 [Amazon ElastiCache 節點申請表](#) 提出請求。

如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 [Amazon ElastiCache 限額](#)。

錯誤訊息：超過客戶節點配額。在此區域中，您最多可以有 %n 個節點 或者，您已達到此區域 %s 個節點的配額。

原因：您嘗試建立或修改某個叢集，結果您帳戶在此區域中跨所有叢集的節點超過 %n 個。

解決方案：變更要求，讓此帳戶在區域中跨所有叢集的節點總數不超過 %n 個。或者，如果您需要超過 %n 個節點，請使用 [Amazon ElastiCache 節點申請表](#) 提出請求。

如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 [Amazon ElastiCache 限額](#)。

錯誤訊息：InsufficientCacheClusterCapacity

原因：AWS 目前隨需容量不足，無法執行您的請求。

解決方案：

- 等候幾分鐘，然後再次提交您的請求；容量會頻繁轉移。
- 以降低的節點或碎片 (節點群組) 數提交新請求。例如，如果您要提出啟動 15 個節點的單一請求，請改嘗試提出 3 個啟動 5 個節點的請求，或 15 個啟動 1 個節點的請求。

- 如果啟動的是叢集，請提交新的請求，而不要指定可用區域。
- 如果啟動的是叢集，請使用不同的節點類型 (您可以在後面的階段進行擴展) 來提交新的請求。如需更多詳細資訊，請參閱 [縮放記憶 ElastiCache 體快取](#)。

## 通知

本主題涵蓋您可能會感興趣的 ElastiCache 通知。通知是一種狀態或事件，在大多數的情況下，通常都是暫時性的，只會持續存在直到找到解決方案或實作解決方案。通知通常會有一個開始日期和解決日期，在解決日期之後，通知便不再重要。任何一個通知可能會與您相關，也可能不相關。我們建議一套實作準則，您可以透過遵循此準則來改善叢集的效能。

通知不會宣告新增或經改善的 ElastiCache 特色或功能。

### 一般 ElastiCache 通知

目前沒有非引擎專用的未解決 ElastiCache 通知。

### ElastiCache for Memcached 通知

以下 ElastiCache 通知僅適用於 Memcached 引擎。

ElastiCache for Memcached 專屬通知

- [提醒：Memcached LRU 爬蟲程式造成分段故障](#)

#### 提醒：Memcached LRU 爬蟲程式造成分段故障

**⚠ 提醒日期：**2017 年 2 月 28 日

在某些情況下，您的叢集可能會在 Memcached LRU Crawler 中發生分段故障時呈現不穩定狀態。此為 Memcached 引擎內的問題，並且已存在一段時間。此問題最早是在根據預設啟用 LRU Crawler 的 Memcached 1.4.33 版中出現。

若您發生此問題，我們建議您停用 LRU Crawler，直到修復為止。若要執行此作業，請在命令列使用 `lru_crawler disable`，或是修改 `lru_crawler` 參數值 (建議)。

解決日期：

解決方案：



# 文檔歷史

- API 版本：2015-02-02
- 文件最新更新時間：2023 年 11 月 27 日

下表說明在 2018 年 3 月之後適用於 中的重要變更。如需有關此文件更新的通知，您可以訂閱 RSS 摘要。

## 最近 ElastiCache 的內存緩存更新

變更	描述	日期
<a href="#">Support 快取 ElastiCache 記憶體 1.6.22</a>	ElastiCache 對於內存緩存現在支持內存緩存 1.6.22。其中包括來自 <a href="#">1.6.18</a> 版的累積錯誤修正。如需詳細資訊，請參閱 <a href="#">Memcached 1.6.22 版</a> 。	2024 年 1 月 10 日
<a href="#">ElastiCache 對於 Memcached 現在支持創建無服務器緩存</a>	您現在可以建立無伺服器快取，它可簡化快取管理並立即擴展，以支援需求最嚴苛的應用程式。如需詳細資訊，請參閱 <a href="#">選擇部署選項</a> 。 <a href="#">新許可</a> 已隨此功能新增至 ElastiCacheServiceRolePolicy 和 AmazonElastiCacheFullAccess，以允許無伺服器快取與受管 VPC 端點建立關聯。此外，已新增許可來支援改良後使用 AmazonElastiCacheFullAccess 政策的主控制台體驗。	2023 年 11 月 27 日
<a href="#">Support 記憶體 ElastiCache 體快取 1.6.17</a>	ElastiCache 對於內存緩存現在支持內存緩存 1.6.17。其中包括 <a href="#">Memcached 1.6.12</a> 版的累	2023 年 1 月 18 日

積錯誤修正。如需詳細資訊，請參閱 [Memcached 1.6.17 版](#)。

### [ElastiCache 對於內存緩存現在支持 IPV6](#)

ElastiCache 支援網際網路通訊協定第 4 版和第 6 版 (IPv4 和 IPv6)，可讓您將叢集設定為只接受 IPv4 連線、僅接受 IPv6 連線，或同時接受 IPv4 和 IPv6 連線 (雙堆疊)。在 [Nitro 系統](#) 上建置的所有執行個體中，使用 Memcached 引擎 1.6.6 版以上的工作負載都支援 IPv6。ElastiCache 透過 IPv6 存取不會產生額外費用。如需詳細資訊，請參閱 [選擇網路類型](#)。

2022 年 11 月 7 日

### [ElastiCache 對於 Memcached 現在支持傳輸中加密](#)

傳輸中加密是選用的功能，可讓您在資料最易遭受攻擊的點，也就是資料在地點間傳輸時，增加資料的安全性。Memcached 1.6.12 和更新版本支援。如需詳細資訊，請參閱 [ElastiCache 傳輸中加密 \(TLS\)](#)。

2022 年 5 月 26 日

## [ElastiCache 現在支持 PrivateLink](#)

AWS PrivateLink 允許您在沒有網際網路閘道、NAT 裝置、VPN 連線或 AWS 直 Connect 連線的情況下私密存取 ElastiCache API 作業。如需詳細資訊，請參閱適用於 Redis 或 [Amazon ElastiCache API 的 Amazon API 和接口虛擬私人雲端端點 \(AWS PrivateLink\)](#) 和 [介面虛擬私人雲端端點 \(AWS PrivateLink\)](#)。ElastiCache

2022 年 1 月 24 日

## [支援標記資源和條件索引鍵](#)

ElastiCache 現在支援標記，協助您管理叢集和其他 ElastiCache 資源。如需詳細資訊，請參閱 [標記 ElastiCache 資源](#)。ElastiCache 還引入了對條件鍵的支持。您可以指定條件，決定 IAM 政策的生效方式。如需這些條件索引鍵的詳細資訊，請參閱 [使用條件索引鍵](#)。

2021 年 4 月 7 日

## [支援 Memcached 1.6.6。](#)

ElastiCache 對於內存緩存現在支持內存緩存 1.6.6。其中包括 Memcached 1.5.16 版的累積錯誤修正。如需詳細資訊，請參閱 [Memcached 1.6.6 版](#)。

2020 年 11 月 12 日

## [ElastiCache 現在可在 AWS Outposts 上使用](#)

[AWS Outposts](#) 將原生 AWS 服務、基礎架構和作業模型帶到幾乎任何資料中心、主機代管空間或內部部署設施。您可以在 Outposts ElastiCache 上進行部署，以便在內部部署設定、操作和使用快取，就像在雲端中一樣。如需詳細資訊，請針對 Redis 參閱[使用 Outposts](#)，或為 Memcached 參考[使用 Outposts](#)。

2020 年 10 月 8 日

## [ElastiCache 現在支持 Local Zones](#)

「本地區域」是地理位置靠近您的使用者的「AWS 區域」的延伸。您可以透過建立新的子網路並將其指派給 Local Zones，將任何虛擬私有雲端 (VPC) 從父 AWS 區域延伸到本機區域。如需詳細資訊，請參閱[使用本地區域](#)。

2020 年 9 月 25 日

## [ElastiCache 現在支援資源層級權限](#)

您現在可以透過在 AWS Identity and Access Management (IAM) 政策中指定 ElastiCache 資源來限制使用者許可範圍。如需詳細資訊，請參閱[資源層級許可](#)。

2020 年 8 月 12 日

## [ElastiCache 現在支援叢集 ElastiCache 的自動更新](#)

Amazon ElastiCache 現在支援在服務更新的「建議套用日期」過後自動更新 ElastiCache 叢集。ElastiCache 使用您的維護時段來排程適用叢集的自動更新。如需詳細資訊，請參閱[自助更新](#)。

2020 年 5 月 13 日

### [Amazon ElastiCache 現在支持 T3 標準緩存節點](#)

您現在可以在 Amazon 中啟動下一代一般用途高載 T3 標準快取節點。ElastiCache Amazon EC2 的 T3 標準執行個體提供 CPU 基準效能，能夠讓 CPU 使用率隨時火力全開，直至累積的額度耗盡。如需詳細資訊，請參閱[支援的節點類型](#)。

2019 年 11 月 12 日

### [ElastiCache Memcached 現在允許用戶按照自己的時間表應用服務更新](#)

有了這項功能，您就能夠選擇在您所選的時間 (而不只是在維護時段期間) 套用可用的服務更新。這個功能可將服務中斷的狀況降到最低 (特別是在業務流程的尖峰期間)，並協助確保您保持符合安全性更新。如需詳細資訊，請參閱 [Amazon 中的自助服務更新 ElastiCache](#)。

2019 年 10 月 9 日

### [Support 快取 ElastiCache 記憶體 1.5.16](#)

ElastiCache 對於內存緩存現在支持內存緩存 1.5.16。其中包括版本 [Memcached 1.5.14](#) 和 [Memcached 1.5.15](#) 的累積錯誤修正。如需詳細資訊，請參閱 [Memcached 1.5.16 版](#)。

2019 年 9 月 6 日

### [ElastiCache 標準預留執行個體供應項目：部分預付、全部預付和無預付。](#)

預留執行個體可讓您根據 ElastiCache 執行個體類型和 AWS 區域靈活地預留一年或三年期的 Amazon 執行個體。如需詳細資訊，請參閱使用[預留節點管理成本](#)。

2019 年 1 月 18 日

## [Support 記憶 ElastiCache 體快 取 1.5.10](#)

ElastiCache 對於內存緩存現在支持內存緩存 1.5.10。它也包含對 no\_modern 和 inline\_ascii\_resp 參數的支援。如需詳細資訊，請參閱 [Memcached 1.5.10 版](#)。

2018 年 11 月 14 日

## [使用者指南重組](#)

單一 ElastiCache 使用者指南現已重新架構，因此 Redis (適用於 Redis 使用者指南) 和 Memcached 的使用者指南 ([ElastiCache 適用於 Memcached 使用者指南](#)) 各有不同的使 [ElastiCache 用者指南](#)。AWS CLI 命令參考：[彈性疼痛區段](#) 和 [Amazon ElastiCache API 參考](#) 中的文件結構保持不變。

2018 年 4 月 20 日

下表說明在 2018 年 3 月之前 ElastiCache 對 使用者指南的 Memcached 使用者指南 ElastiCache 進行的重要變更。

變更	描述	變更日期
支援亞太區域 (大阪—當地)。	<p>ElastiCache 增加了對亞太地區 (大阪本地) 區域的支持。亞太區域 (大阪) 目前支援單一可用區域，且僅能透過邀請來取得。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li><a href="#">支援的區域</a></li> <li><a href="#">支援的快取節點類型</a></li> </ul>	2018 年 2 月 12 日
支援歐洲 (巴黎)。	<p>ElastiCache 增加了對歐盟 (巴黎) 區域的支持。如需詳細資訊，請參閱下列內容：</p>	2017 年 12 月 18 日

變更	描述	變更日期
	<ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	
支援中國 (寧夏) 區域	<p>Amazon ElastiCache 增加了對中國 ( 寧夏 ) 地區的支持。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2017 年 12 月 11 日
支援服務連結角色	<p>此版本對服務連結角色 (SLR) 的 ElastiCache 新增支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">為 Amazon ElastiCache 使用服務連結角色</a></li> <li>• <a href="#">設定您的權限 (僅限新 ElastiCache 使用者)</a></li> </ul>	2017 年 12 月 7 日
支援 R4 節點類型	<p>此版本在所有支援的 AWS 區域中 ElastiCache 新增支援 R4 節點類型。ElastiCache 您能以隨需或預留快取節點形式購買 R4 節點類型。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的快取節點類型</a></li> <li>• <a href="#">Memcached 節點類型專用參數</a></li> </ul>	2017 年 11 月 20 日
連接模式主題	<p>ElastiCache 文件新增一個主題，內容涵蓋在 Amazon VPC 中存取 ElastiCache 叢集各種模式。</p> <p>如需詳細資訊，請參閱《ElastiCache 使用者指南》中的 <a href="#">在 Amazon VPC 中存取 ElastiCache 快取的存取模式</a>。</p>	2017 年 4 月 24 日

變更	描述	變更日期
支援 Memcached 1.4.34	<p>ElastiCache 增加了支援 Memcached 版本 1.4.34，其中包含了一些修復到早期的 Memcached 版本。</p> <p>如需詳細資訊，請參閱上的 <a href="#">Memcached 1.4.34 版本說明</a>。GitHub</p>	2017 年 4 月 10 日
支援 Memcached 1.4.33	<p>ElastiCache 增加了對內存緩存版本 1.4.33 的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">Memcached 1.4.33 版</a></li> <li>• <a href="#">Memcached 1.4.33 版新增參數</a></li> </ul>	2016 年 12 月 20 日
支援西歐 (倫敦) 區域	<p>ElastiCache 增加了對歐盟 (倫敦) 區域的支援。目前只支援 T2 和 M4 節點類型。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2016 年 12 月 13 日
支援加拿大 (蒙特婁) 區域	<p>ElastiCache 增加了對加拿大 (蒙特利爾) 區域的支援。此 AWS 區域目前僅支援節點類型 M4 和 T2。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2016 年 12 月 8 日
支援 M4 和 R3 節點類型	<p>ElastiCache 增加了對南美洲 (聖保羅) 區域 R3 和 M4 節點類型的支援，以及中國 (北京) 區域的 M4 節點類型。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2016 年 11 月 1 日

變更	描述	變更日期
美國東部 2 (俄亥俄) 區域支援	<p>ElastiCache 使用 M4、T2 和 R3 節點類型增加了對美國東部 (俄亥俄) 區域 (US-東部 -2) 的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的區域</a></li> <li>• <a href="#">支援的快取節點類型</a></li> </ul>	2016 年 10 月 17 日
支援 M4 節點類型	<p>ElastiCache 在大多數支援的 AWS 區域中加入對 M4 節點類型系列的支援。您能以隨需或預留快取節點形式購買 M4 節點類型。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的快取節點類型</a></li> <li>• <a href="#">Memcached 節點類型專用參數</a></li> </ul>	2016 年 8 月 3 日
支援孟買地區	<p>ElastiCache 新增對亞太區域 (孟買) 區域的支援。如需詳細資訊，請參閱下列內容：</p> <ul style="list-style-type: none"> <li>• <a href="#">支援的快取節點類型</a></li> <li>• <a href="#">Memcached 節點類型專用參數</a></li> </ul>	2016 年 6 月 27 日
支援 R3 節點類型	<p>ElastiCache 在中國 (北京) 區域和南美洲 (聖保羅) 區域增加了對 R3 節點類型的支援。如需詳細資訊，請參閱<a href="#">支援的快取節點類型</a>。</p>	2016 年 3 月 16 日
ElastiCache 使用 Lambda 函數存取	<p>已新增設定 Lambda 函數以在 Amazon VPC ElastiCache 中存取的教學課程。如需詳細資訊，請參閱 <a href="#">ElastiCache 教學和影片</a>。</p>	2016 年 2 月 12 日
支援亞太區域 (首爾)	<p>ElastiCache 增加了對具有 t2，m3 和 r3 節點類型的亞太區域 (首爾) ( AP-東北 -2 ) 區域的支持。</p>	2016 年 1 月 6 日

變更	描述	變更日期
支援 Memcached 1.4.28。	ElastiCache 增加了對自 1.4.14 版本以來的內存緩存版本 1.4.24 和內存緩存改進的支持。此版本新增了最久未使用 (LRU) 快取管理的背景工作、jenkins 或 murmur3 雜湊演算法選項、新命令以及其他錯誤修正。如需詳細資訊，請參閱 <a href="#">Memcached 版本備註</a> 。	2015 年 8 月 27 日
支援使用 PHP 5.6 的 Memcached 自動探索	此版本的 Amazon ElastiCache 增加了對於 PHP 5.6 版本的 Memcached 自動發現客戶端的支持。如需詳細資訊，請參閱 <a href="#">編譯適用於 PHP 的 ElastiCache 叢集用戶端原始碼</a> 。	2015 年 7 月 29 日
新主題：ElastiCache 從外部訪問 AWS	已新增有關如何從外部存取 ElastiCache 資源的新主題 AWS。如需詳細資訊，請參閱 <a href="#">ElastiCache 從外部存取 AWS</a> 。	2015 年 7 月 9 日
新增節點取代訊息	ElastiCache 新增三個與排程節點取代相關的訊息 ElastiCacheElastiCache : NodeReplacementScheduledNodeReplacementRescheduled、ElastiCache:和 : NodeReplacementCanceled。  如需排程要取代節點時可採取的詳細資訊和動作，請參閱 ElastiCache的 <a href="#">事件通知</a> 和 <a href="#">Amazon SNS</a> 。	2015 年 6 月 11 日
支援成本分配標籤	ElastiCache 新增對成本分配標籤的支援。如需詳細資訊，請參閱 <a href="#">使用成本配置標籤監控成本</a> 。	2015 年 2 月 9 日
Support AWS GovCloud (美國西部) 區域	ElastiCache 增加對 AWS GovCloud (美國西部) (us-gov-west-1) 區域的支援。	2015 年 1 月 29 日
支援歐洲 (法蘭克福) 區域	ElastiCache 增加了對歐洲 (法蘭克福) ( 歐盟中部 -1 ) 區域的支持。	2015 年 1 月 19 日

變更	描述	變更日期
AWS CloudTrail 支援記錄 API 呼叫	ElastiCache 添加了對使用 AWS CloudTrail 記錄所有 ElastiCache API 調用的支持。如需詳細資訊，請參閱 <a href="#">使用 AWS CloudTrail 記錄 Amazon ElastiCache API 呼叫</a> 。	2014 年 9 月 15 日
支援新的執行個體大小	ElastiCache 新增額外一般用途 (T2) 執行個體的支持。如需詳細資訊，請參閱 <a href="#">使用參數群組設定引擎參數</a> 。	2014 年 9 月 11 日
Memcached 支援更靈活的節點置放	ElastiCache 添加了跨多個可用區域創建 Memcached 節點的支持。	2014 年 7 月 23 日
支援新的執行個體大小	ElastiCache 新增額外一般用途 (M3) 執行個體和記憶體最佳化 (R3) 執行個體的支持。如需詳細資訊，請參閱 <a href="#">使用參數群組設定引擎參數</a> 。	2014 年 7 月 1 日
PHP 自動探索	已新增對 PHP 5.5 版自動探索的支援。	2014 年 5 月 13 日
支援預設 Amazon Virtual Private Cloud (VPC)	在此版本中，ElastiCache 已與 Amazon Virtual Private Cloud (VPC) 完全整合。若是新客戶，則快取叢集預設會在 Amazon VPC 中建立。如需詳細資訊，請參閱 <a href="#">Amazon VPC 和 ElastiCache 安全性</a> 。	2013 年 1 月 8 日
PHP 支援快取節點自動探索	快取節點自動探索的初始版本已提供對 Java 程式的支援。在此版本中，ElastiCache 為 PHP 帶來快取節點 auto 探索支援。	2013 年 1 月 2 日
支援 Amazon Virtual Private Cloud (VPC)	在此版本中，ElastiCache 叢集可以在 Amazon Virtual Private Cloud (VPC) 中啟動。根據預設，新客戶的快取叢集會自動在 Amazon VPC 中建立；現有的客戶則可按自己的步調遷移至 Amazon VPC。如需詳細資訊，請參閱 <a href="#">Amazon VPC 和 ElastiCache 安全性</a> 。	2012 年 12 月 20 日

變更	描述	變更日期
快取節點自動探索和新的快取引擎版本	<p>ElastiCache 提供快取節點 auto 動探索功能 — 用戶端程式可自動判斷叢集中所有快取節點，以及啟動和維護與所有這些節點的連線。</p> <p>此版本也提供新的快取引擎版本：Memcached 1.4.14 版。這款新的快取引擎提供增強型 slab 重新平衡功能、效能和可擴展性大幅提升，以及多項錯誤修正。您可以設定多種新的快取參數。如需詳細資訊，請參閱 <a href="#">使用參數群組設定引擎參數</a>。</p>	2012 年 11 月 28 日
新增快取節點類型	本版本提供四種額外的快取節點類型。	2012 年 11 月 13 日
預留快取節點	此版本新增對預留快取節點的支援。	2012 年 4 月 5 日
新指南	這是 Amazon ElastiCache 用戶指南的第一個版本。	2011 年 8 月 22 日

# AWS 詞彙表

有關最新 AWS 術語，請參閱AWS 詞彙表 參考文獻中的[AWS 詞彙表](#)。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。