Application Load Balancer

Elastic Load Balancing

Elastic Load Balancing: Application Load Balancer

Table of Contents

什麼是 Application Load Balancer？ 1
Application Load Balancer 元件 1
Application Load Balancer 概觀 2
從 Classic Load Balancer 遷移的優點 2
相關服務 3
定價 4
Application Load Balancer 5
負載平衡器的子網路 6
可用區域子網路 6
Local Zone 子網路 7
Outpost 子網路 7
負載平衡器安全群組 9
負載平衡器狀態 9
負載平衡器屬性 9
IP 地址類型 12
Application Load Balancer IP 地址管理 13
IPAM IP 地址集區 13
負載平衡器連線 14
跨區域負載平衡 14
DNS 名稱 14
建立負載平衡器 15
先決條件 16
建立負載平衡器 16
測試負載平衡器 20
後續步驟 21
更新可用區域 21
更新安全群組 22
建議的規則 23
更新相關聯的安全群組 25
更新 IP 地址類型 26
更新 IPAM IP 地址集區 27
編輯負載平衡器屬性 29
連線閒置逾時 29
HTTP 用戶端保持連線持續時間 30

刪除保護 32
去同步緩解模式 34
主機標頭保留 36
標記負載平衡器 39
刪除負載平衡器 41
檢視資源映射 42
資源地圖元件 42
區域轉移 43
開始之前 44
跨區域負載平衡 44
管理覆寫 45
啟用區域轉移 45
開始區域轉移 46
更新區域轉移 47
取消區域轉移 48
LCU 保留 49
請求保留 50
更新或取消保留 52
監控保留 52
負載平衡器整合 53
Amazon Application Recovery Controller (ARC) 54
Amazon CloudFront + AWS WAF 54
AWS Global Accelerator 55
AWS Config 55
AWS WAF 55
接聽程式和規則 57
接聽程式組態 57
接聽程式屬性 58
預設動作 60
建立 HTTP 接聽程式 60
先決條件 60
新增 HTTP 接聽程式 61
SSL 憑證 63
預設憑證 64
憑證清單 64
憑證續約 65

安全政策 65
describe-ssl-policies 命令範例 67
TLS 安全政策 68
FIPS 安全政策 97
FS 支援的政策 118
建立 HTTPS 接聽程式 124
先決條件 124
新增 HTTPS 接聽程式 125
更新 HTTPS 接聽程式 127
更換預設憑證 127
將憑證新增至憑證清單 129
從憑證清單中移除憑證 130
更新安全政策 131
HTTP 標頭修改 133
接聽程式規則 133
動作類型 134
條件類型 141
轉換 148
新增規則 150
編輯規則 156
刪除規則 161
交互 TLS 驗證 162
開始之前 163
HTTP 標頭 165
公告 CA 主旨名稱 167
連線日誌 167
設定交互 TLS 167
共用信任存放區 175
使用者身分驗證 179
準備使用 OIDC 合規 IdP 179
準備使用 Amazon Cognito 180
準備使用 Amazon CloudFront 182
設定使用者身分驗證 182
身分驗證流程 185
使用者宣告編碼和簽章驗證 186
Timeout (逾時) 188

身分驗證登出 189
JWT 驗證 189
準備使用 JWT 驗證 190
JWT 驗證限制 190
使用 CLI 設定 JWT 驗證 191
X-Forwarded 標頭 193
X-Forwarded-For 193
X-Forwarded-Proto 197
X-Forwarded-Port 198
HTTP 標頭修改 198
重新命名 mTLS/TLS 標頭 198
新增回應標頭 199
停用標頭 201
限制 201
啟用標頭修改 201
刪除接聽程式 204
目標群組 206
路由組態 207
Target type (目標類型) 207
IP 地址類型 209
通訊協定版本 209
已登記的目標 210
目標最佳化工具 211
目標群組屬性 211
目標群組運作狀態 213
運作運作狀態不佳 214
需求和考量事項 214
監控 215
範例 215
針對您的負載平衡器使用 Route 53 DNS 備援 217
建立目標群組 217
設定運作狀態檢查 220
運作狀態檢查設定 221
目標運作狀態 223
運作狀態檢查原因代碼 224
檢查目標運作狀態 225

更新運作狀態檢查設定 ... 227
編輯目標群組屬性 ... 229
取消登記的延遲 ... 229
路由演算法 ... 230
慢速啟動模式 ... 232
運作狀態設定 ... 234
跨區域負載平衡 ... 236
自動目標權重 (ATW) ... 239
黏性工作階段 ... 242
登記目標 ... 249
目標安全群組 ... 249
目標最佳化工具 ... 250
共用子網路 ... 252
登記目標 ... 252
取消註冊目標 ... 254
使用 Lambda 函數做為目標 ... 255
準備 Lambda 函數 ... 256
為 Lambda 函數建立目標群組 ... 256
從負載平衡器接收事件 ... 258
對負載平衡器進行回應 ... 259
多值標頭 ... 260
啟用運作狀態檢查 ... 263
註冊 Lambda 函數 ... 265
取消註冊 Lambda 函數 ... 266
標記目標群組 ... 267
刪除目標群組 ... 269
監控負載平衡器 ... 270
CloudWatch 指標 ... 271
Application Load Balancer 指標 ... 271
Application Load Balancer 的指標維度 ... 294
Application Load Balancer 指標的統計資料 ... 294
檢視負載平衡器的 CloudWatch 指標 ... 295
存取日誌 ... 297
存取日誌檔 ... 298
存取日誌項目 ... 299
範例日誌項目 ... 315

設定日誌交付通知 317
處理存取日誌檔 317
啟用存取日誌 318
停用存取日誌 327
連線日誌 328
連線日誌檔案 328
連線日誌項目 330
範例日誌項目 333
處理連線日誌檔案 333
啟用連線日誌 334
停用連線日誌 341
運作狀態檢查日誌 342
運作狀態檢查日誌檔案 343
運作狀態檢查日誌項目 344
範例日誌項目 346
設定日誌交付通知 347
處理運作狀態檢查日誌檔案 347
啟用運作狀態檢查日誌 347
停用運作狀態檢查日誌 355
請求追蹤 355
語法 356
限制 357
為您的負載平衡器進行疑難排解 358
已註冊目標處於非服務中狀態 358
用戶端無法連接到面向網際網路的負載平衡器 359
負載平衡器不會收到傳送至自訂域的請求 360
傳送至負載平衡器的 HTTPS 要求會傳回 "NET::ERR_CERT_COMMON_NAME_INVALID" 360
負載平衡器顯示處理時間延長 361
負載平衡器會傳送 000 的回應代碼 361
負載平衡器產生 HTTP 錯誤 361
HTTP 400：錯誤的請求 362
HTTP 401：未經授權 362
HTTP 403：禁止 363
HTTP 405：方法不允許 363
HTTP 408：請求逾時 363
HTTP 413：承載過大 363

HTTP 414：URI 過長 363
HTTP 460 364
HTTP 463 364
HTTP 464 364
HTTP 500：內部伺服器錯誤 364
HTTP 501：未導入 365
HTTP 502：無效的閘道 365
HTTP 503：服務無法使用 365
HTTP 504：閘道逾時 366
HTTP 505：不支援的版本 366
HTTP 507：儲存不足 366
HTTP 561：未經授權 366
HTTP 562：JWKS 請求失敗 366
目標產生了 HTTP 錯誤 367
AWS Certificate Manager 憑證無法使用 367
不支援多行標頭 367
使用資源映射對運作狀態不佳的目標進行故障診斷 367
針對目標最佳化工具進行故障診斷 369
配額 371
負載平衡器 371
目標群組 372
Rules 372
信任存放區 373
憑證 373
HTTP 標頭 373
Load Balancer容量單位 374
文件歷史紀錄 375
..... ccclxxxi

什麼是 Application Load Balancer？

Elastic Load Balancing 會自動將傳入流量分配到一或多個可用區域中的多個目標，例如 EC2 執行個體、容器和 IP 地址。其會監控已註冊目標的運作狀態，並且僅將流量路由至運作狀態良好的目標。當傳入流量隨著時間發生變化，Elastic Load Balancing 會擴展您的負載平衡器。他可以自動擴展以因應絕大多數的工作負載。

Elastic Load Balancing 支援下列負載平衡器：Application Load Balancer、Network Load Balancer、Gateway Load Balancer 和 Classic Load Balancer。您可以選取最符合您需要的負載平衡器類型。本指南主要介紹 Application Load Balancer。如需有關其他負載平衡器的詳細資訊，請參閱 User Guide for Network Load Balancers、User Guide for Gateway Load Balancers 和 《Classic Load Balancer 使用者指南》。

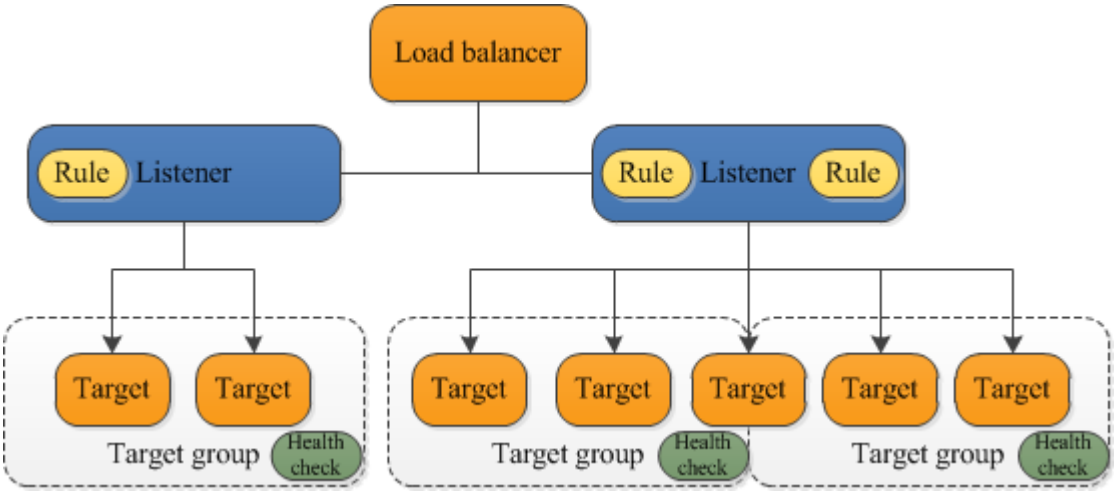
Application Load Balancer 元件

負載平衡器做為用戶端的單一聯絡點。負載平衡器會將傳入的應用程式流量分散到多個可用區域中的多個目標，例如 EC2 執行個體。這會提高您應用程式的可用性。您要為負載平衡器添加一個或多個接聽程式。

接聽程式會使用您所設定的通訊協定與連接埠，檢查來自用戶端的連線請求。您為接聽程式定義的規則，將決定負載平衡器路由請求到已登錄的目標的方法。每個規則由優先順序、一或多個動作及一或多個條件組成。滿足規則的條件時，即會執行它的動作。您必須為每個接聽程式定義預設規則，也可以選擇性地定義額外的規則。

每個目標群組會使用您指定的通訊協定和連接埠號碼，將請求路由至一個或多個已註冊的目標，例如 EC2 執行個體。您可以向多個目標群組註冊任一目標。您可以針對每個目標群組設定運作狀態檢查。凡已註冊至負載平衡器的接聽程式規則中指定之目標群組的所有目標，系統將對其執行運作狀態檢查。

下圖說明基本元件。請注意，每個接聽程式包含預設規則，而一個接聽程式包含另一個規則，可將請求路由至不同的目標群組。一個目標向兩個目標群組註冊。



如需詳細資訊，請參閱下列 文件：

- 負載平衡器
- 接聽程式
- 目標群組

Application Load Balancer 概觀

Application Load Balancer 在應用程式層 (開放系統互連 (OSI) 模型的第七層) 運作。當負載平衡器收到請求後，它會依優先順序評估接聽程式規則，以決定要套用的規則，然後從目標群組中選取規則動作的目標。您可以設定接聽程式規則，以根據應用程式流量的內容，將請求路由到不同的目標群組。即使一個目標向多個目標群組註冊，每個目標群組的路由都是獨立運作。您可以設定在目標群組層級上使用的路由演算法。預設路由演算法是循環式；或者，您可以指定最少未完成的請求路由演算法。

您可以依據需求變更，為負載平衡器新增和移除目標，而不會中斷應用程式整體的請求流程。當應用程式的流量隨著時間發生變化，Elastic Load Balancing 會擴展您的負載平衡器。Elastic Load Balancing 能夠自動擴展以因應絕大多數的工作負載。

您可以設定運作狀態檢查，用於監控已註冊目標的運作狀態，使負載平衡器只能傳送請求至運作狀態良好的目標。

如需詳細資訊，請參閱 Elastic Load Balancing 使用者指南中的 Elastic Load Balancing的運作方式 。

從 Classic Load Balancer 遷移的優點

使用 Application Load Balancer (而非 Classic Load Balancer) 具有下列優點：

- 支援 路徑條件。您可以為接聽程式設定規則，以根據請求中的 URL 來轉送請求。這可讓您將應用程式建構成較小的服務，以根據 URL 的內容，將請求路由傳送到正確的服務。
- 支援 主機條件。您可以為接聽程式設定規則，以根據 HTTP 標頭中的主機欄位來轉送請求。這可讓您使用單一負載平衡器，將請求路由至多個網域。
- 支援根據請求中的欄位 (例如 HTTP 標頭條件 和方法、查詢參數及來源 IP 地址) 來路由。
- 支援將請求路由至單一 EC2 執行個體上的多個應用程式。您可將一個執行個體或 IP 地址註冊到多個目標群組，每個目標群組都在不同的連接埠上。
- 支援將請求從一個 URL 重新導向另一個 URL。
- 支援傳回自訂 HTTP 回應。
- 支援透過 IP 地址註冊目標，包括位於負載平衡器的 VPC 外部的目標。
- 支援註冊 Lambda 函數做為目標。
- 支援負載平衡器在路由請求之前，透過企業或社交身分來驗證應用程式的使用者。
- 支援容器化的應用程式。Amazon Elastic Container Service (Amazon ECS) 可在排程任務時選取未使用的連接埠，並使用此連接埠向目標群組註冊該任務。這使您得以有效利用您的叢集。
- 支援單獨監控各項服務的運作狀態，因為運作狀態檢查是在目標群組層級設定，而許多 CloudWatch 指標的回報也是在目標群組層級。將目標群組連接到 Auto Scaling 群組令您能夠隨需動態擴展各項服務。
- 存取日誌包含其他資訊，且以壓縮格式存放。
- 提升負載平衡器的效能。

如需每個負載平衡器類型支援之功能的詳細資訊，請參閱 Elastic Load Balancing 功能。

相關服務

Elastic Load Balancing 適用以下服務，可改善應用程式的可用性和可擴展性。

- Amazon EC2 – 在雲端執行應用程式的虛擬伺服器。您可以設定負載平衡器，將流量路由到 EC2 執行個體。
- Amazon EC2 Auto Scaling – 確保您正在執行所需數量的執行個體 (即使其中某個執行個體處於故障狀態)，並可讓您隨著執行個體需求的變更，自動增加或減少執行個體的數量。如果您啟用具有 Elastic Load Balancing 的 Auto Scaling，Auto Scaling 啟動的執行個體會自動在目標群組中註冊，而由 Auto Scaling 終止的執行個體會自動從目標群組中取消註冊。

- AWS Certificate Manager – 建立 HTTPS 接聽程式時，可以指定 ACM 所提供的憑證。負載平衡器會使用此憑證來終止連線，並解密來自用戶端的請求。如需詳細資訊，請參閱Application Load Balancer 的 SSL 憑證。
- Amazon CloudWatch – 可讓您監控負載平衡器並視需要來採取動作。如需詳細資訊，請參閱適用於 Application Load Balancer 的 CloudWatch 指標。
- Amazon ECS – 可讓您在 EC2 執行個體叢集上執行、停止和管理 Docker 容器。您可以設定負載平衡器，將流量路由到容器。如需詳細資訊，請參閱 Amazon Elastic Container Service Developer Guide 中的 Service load balancing。
- AWS Global Accelerator – 改善應用程式的可用性和效能。使用 加速器將流量分配到一或多個 AWS 區域中的多個負載平衡器。如需詳細資訊，請參閱《AWS Global Accelerator 開發人員指南》https://docs.aws.amazon.com/global-accelerator/latest/dg/。
- Route 53 – 透過將網域名稱 (如 `www.example.com`) 轉換為電腦用來互相連線的數字 IP 地址 (例如 `192.0.2.1`)，提供可靠且經濟實惠的方式來將訪客路由至網站。 AWS 會將 URL 指派至負載平衡器等資源。不過，您可能需要能讓使用者輕鬆記住的 URL。例如，您可以將網域名稱映射至負載平衡器。如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的將流量路由到 ELB 負載平衡器。
- AWS WAF — 您可以使用 AWS WAF Application Load Balancer 根據 Web 存取控制清單 (Web ACL) 中的規則來允許或封鎖請求。如需詳細資訊，請參閱AWS WAF。

若要檢視與您的負載平衡器整合之服務的相關資訊，請在 中選取您的負載平衡器 AWS 管理主控台 ，然後選擇整合式服務索引標籤。

定價

使用負載平衡器時，您只需按實際用量付費。如需詳細資訊，請參閱 Elastic Load Balancing 定價。

Application Load Balancer

負載平衡器做為用戶端的單一聯絡點。用戶端將請求傳送到負載平衡器，而負載平衡器將請求傳送到如 EC2 執行個體等的目標。若要設定您的負載平衡器，您需要建立目標群組，然後使用您的目標群組來登錄目標。您也可以建立接聽程式，以檢查來自用戶端的連線請求，並建立接聽程式規則，將來自用戶端的請求路由到一或多個目標群組中的目標。

如需詳細資訊，請參閱 Elastic Load Balancing 使用者指南中的 Elastic Load Balancing的運作方式 。

目錄

- 負載平衡器的子網路
- 負載平衡器安全群組
- 負載平衡器狀態
- 負載平衡器屬性
- IP 地址類型
- Application Load Balancer IP 地址管理
- IPAM IP 地址集區
- 負載平衡器連線
- 跨區域負載平衡
- DNS 名稱
- 建立 Application Load Balancer
- 更新 Application Load Balancer 的可用區域
- Application Load Balancer 的安全群組
- 更新 Application Load Balancer 的 IP 地址類型
- 更新 Application Load Balancer 的 IPAM IP 地址集區
- 編輯 Application Load Balancer 的屬性
- 標記 Application Load Balancer
- 刪除 Application Load Balancer
- 檢視 Application Load Balancer 資源映射
- Application Load Balancer 的區域轉移

- Application Load Balancer 的容量保留
- Application Load Balancer 的整合

負載平衡器的子網路

建立 Application Load Balancer 時，必須啟用內含目標的區域。若要啟用區域，請在該區域中指定子網路。Elastic Load Balancing 會在您指定的每個區域建立負載平衡器節點。

考量事項

- 每個已啟用的區域擁有至少一個已註冊的目標時，負載平衡器的效率最高。
- 如果您在某個區域內註冊目標但未啟用該區域，這些已註冊的目標便不會接收來自負載平衡器的流量。
- 如果您為負載平衡器啟用多個區域，這些區域必須是相同類型。例如，您無法同時啟用可用區域和 Local Zone。
- 您可以指定與您共用的子網路。
- Elastic Load Balancing 會在您設定負載平衡器的子網路中建立網路介面。這些網路介面會保留，因此即使子網路在可用 IP 地址上執行不足，負載平衡器仍可完成維護動作。它們具有描述 "ENI reserved by ELB for subnet"。

Application Load Balancers 支援以下子網路類型。

子網類型

- 可用區域子網路
- Local Zone 子網路
- Outpost 子網路

可用區域子網路

您必須選取至少兩個可用區域子網路。適用以下限制：

- 每個子網路都必須來自不同的可用區域。
- 為了確保負載平衡器可以適當調整規模，請確認負載平衡器的每個可用區域子網路有一個 CIDR 區塊，並具有至少一個 /27 位元遮罩 (例如，10.0.0.0/27)，且每個子網路至少有八個可用 IP 地址。需要有八個可用 IP 地址，才能讓負載平衡器視需要橫向擴展。負載平衡器會使用這些 IP 地址與

目標建立連線。如果沒有這些地址，Application Load Balancer 可能會在嘗試取代節點時遇到困難，而導致其變成失敗狀態。

注意：如果嘗試擴展時，Application Load Balancer 子網路用完可用的 IP 地址，Application Load Balancer 將在容量不足的情況下執行。在此期間，舊節點會繼續為流量提供服務，但在嘗試建立連線時，停滯的擴展嘗試可能會導致 5xx 錯誤或逾時。

Local Zone 子網路

您可以指定 Local Zone 子網路。本機區域子網路不支援下列功能：

- Lambda 函數作為目標
- 交互 TLS 驗證
- AWS WAF 整合

Outpost 子網路

您可以指定單一的 Outpost 子網路。適用以下限制：

- 內部部署資料中心必須已安裝和設定 Outpost。Outpost 與其 AWS 區域之間必須有可靠的網路連線。如需詳細資訊，請參閱「AWS Outposts 使用者指南」。
- 負載平衡器在負載平衡器節點的 Outpost 上需要兩個 `large` 執行個體。下表所示是支援的執行個體類型。負載平衡器會視需要擴展，一次調整一個節點的大小 (從 `large` 到 `xlarge`，然後從 `xlarge` 到 `2xlarge`，接著從 `2xlarge` 到 `4xlarge`)。將節點擴展至最大的執行個體大小之後，如果您需要額外的容量，負載平衡器會將 `4xlarge` 執行個體新增為負載平衡器節點。如果您沒有足夠的執行個體容量或可用的 IP 地址來擴展負載平衡器，負載平衡器會向 AWS Health 儀板表 報告事件，且負載平衡器狀態會變為 `active_impaired`。
- 您可以依執行個體 ID 或 IP 地址來註冊目標。如果您在 Outpost 的 AWS 區域中註冊目標，則不會使用這些目標。
- 不支援以下 功能：
  - AWS Global Accelerator 整合
  - Lambda 函數作為目標
  - 交互 TLS 驗證
  - 黏性工作階段
  - 使用者身分驗證

- AWS WAF 整合

Application Load Balancer 可部署在 Outpost 的 c5/c5d、m5/m5d 或 r5/r5d 執行個體上。下表顯示負載平衡器可在 Outpost 上使用的每個執行個體類型的大小和 EBS 磁碟區：

執行個體類型和大小	EBS 磁碟區 (GB)
c5/c5d	
大型	50
xlarge	50
2xlarge	50
4xlarge	100
m5/m5d	
大型	50
xlarge	50
2xlarge	100
4xlarge	100
r5/r5d	
大型	50
xlarge	100
2xlarge	100
4xlarge	100

負載平衡器安全群組

安全群組扮演防火牆的角色，可控制允許進出負載平衡器的流量。您可以選擇連接埠和通訊協定，以同時允許傳入和傳出流量。

與負載平衡器相關聯之安全群組的規則，在接聽程式連接埠和運作狀態檢查連接埠上都必須允許這兩個方向的流量。當您將接聽程式新增至負載平衡器，或更新目標群組的運作狀態檢查連接埠時，您必須檢閱安全群組規則，以確保它們在新的連接埠上同時允許這兩個方向的流量。如需詳細資訊，請參閱建議的規則。

負載平衡器狀態

負載平衡器可以是以下其中一個狀態：

`provisioning`

正在設定負載平衡器。

`active`

負載平衡器已設定完成並準備好路由流量。

`active_impaired`

負載平衡器正在路由流量，但不具備擴展所需的資源。

`failed`

無法設定的負載平衡器。

負載平衡器屬性

您可以透過編輯 Application Load Balancer 的屬性來設定 Application Load Balancer。如需詳細資訊，請參閱編輯負載平衡器屬性。

以下是負載平衡器屬性：

`access_logs.s3.enabled`

指出在 Amazon S3 中存放的存取日誌是否啟用。預設值為 `false`。

`access_logs.s3.bucket`

存取日誌的 Amazon S3 儲存貯體名稱。如果啟用存取日誌，則此為必要屬性。如需詳細資訊，請參閱啟用存取日誌。

`access_logs.s3.prefix`

Amazon S3 儲存貯體中的位置字首。

`client_keep_alive.seconds`

用戶端保持連線值，以秒為單位。預設值為 3600 秒。

`deletion_protection.enabled`

表示是否已啟用刪除保護。預設值為 `false`。

`idle_timeout.timeout_seconds`

閒置逾時值 (以秒為單位)。預設值為 60 秒。

`ipv6.deny_all_igw_traffic`

封鎖網際網路閘道 (IGW) 對負載平衡器的存取，以防止透過網際網路閘道對內部負載平衡器進行非預期存取。如果是面向網際網路的負載平衡器，設為 `false`，如果是內部負載平衡器，則設為 `true`。此屬性不會阻止非 IGW 網際網路存取 （ 例如透過對等互連、Transit Gateway AWS Direct Connect、 或 Site-to-Site VPN)。

`routing.http.desync_mitigation_mode`

決定負載平衡器如何處理可能對應用程式造成安全風險的請求。可能的值為 `monitor`、`defensive` 和 `strictest`。預設值為 `defensive`。

`routing.http.drop_invalid_header_fields.enabled`

指出具有無效標頭欄位的 HTTP 標頭是否已被負載平衡器 (`true`) 移除或已路由至目標 (`false`)。預設值為 `false`。如 HTTP 欄位名稱登錄檔中所述，Elastic Load Balancing 會要求有效的 HTTP 標頭名稱符合規則運算式 `[-A-Za-z0-9]+`。每個名稱由英數字元或連字號組成。如果要從請求中移除不符合此模式的 HTTP 標頭，請選取 `true`。

`routing.http.preserve_host_header.enabled`

指示 Application Load Balancer 是否應保留 HTTP 請求中的 `Host` 標頭，並將該標頭傳送到目標而不進行任何變更。可能的值為 `true` 和 `false`。預設值為 `false`。

`routing.http.x_amzn_tls_version_and_cipher_suite.enabled`

指示在向目標傳送用戶端請求前，是否在用戶端請求中新增這兩個標頭 (`x-amzn-tls-version` 和 `x-amzn-tls-cipher-suite`) (內含與交涉 TLS 版本和密碼套件相關的資訊)。`x-amzn-tls-version` 標頭包含有關與用戶端交涉的 TLS 通訊協定版本的資訊，而 `x-amzn-tls-cipher-suite` 標頭包含有關與用戶端交涉的密碼套件的資訊。兩個標頭都採用 OpenSSL 格式。此屬性的可能值為 `true` 和 `false`。預設值為 `false`。

`routing.http.xff_client_port.enabled`

指示 `X-Forwarded-For` 標頭是否應該保留用戶端用來連線到負載平衡器的來源連接埠。可能的值為 `true` 和 `false`。預設值為 `false`。

`routing.http.xff_header_processing.mode`

可讓您在 Application Load Balancer 將 HTTP 請求傳送至目標之前修改、保留或移除該請求中的 `X-Forwarded-For` 標頭。可能的值為 `append`、`preserve` 和 `remove`。預設值為 `append`。

- 如果該值為 `append`，Application Load Balancer 會在將 HTTP 請求傳送至目標之前將 HTTP 請求中的 (最近一次跳轉的) 用戶端 IP 地址新增至該請求中的 `X-Forwarded-For` 標頭。
- 如果該值為 `preserve`，Application Load Balancer 會保留 HTTP 請求中的 `X-Forwarded-For` 標頭，並將該請求傳送到目標，而不進行任何變更。
- 如果該值為 `remove`，Application Load Balancer 會在將 HTTP 請求傳送至目標之前移除該請求中的標頭 `X-Forwarded-For`。

`routing.http2.enabled`

指出用戶端是否可以使用 HTTP/2 連線到負載平衡器。如果 `true`，用戶端可以使用 HTTP/2 或 HTTP/1.1 進行連線。如果為 `false`，用戶端必須使用 HTTP/1.1 進行連線。預設值為 `true`。

`waf.fail_open.enabled`

指出是否允許 AWS WAF已啟用負載平衡器在無法轉送請求時，將請求路由至目標 AWS WAF。可能的值為 `true` 和 `false`。預設值為 `false`。

> **Note**
>
> 引入了 `routing.http.drop_invalid_header_fields.enabled` 屬性，可提供 HTTP 去同步保護。新增了 `routing.http.desync_mitigation_mode` 屬性，可為應用程式提供更全面的 HTTP 去同步保護。您不需要同時使用這兩個屬性，並且可以選擇最符合您應用程式需求的屬性。

IP 地址類型

您可以設定 IP 地址類型，用戶端可以使用此類型的 IP 地址存取面向網際網路的負載平衡器和內部負載平衡器。

Application Load Balancer 支援下列 IP 地址類型：

ipv4

用戶端必須採用 IPv4 地址 (例如，192.0.2.1) 才能連接負載平衡器。

dualstack

用戶端可以使用 IPv4 地址 (例如，192.0.2.1) 和 IPv6 地址 (例如，2001:0db8:85a3:0:0:8a2e:0370:7334) 連接至負載平衡器。

dualstack-without-public-ipv4

用戶端必須使用 IPv6 地址（例如 2001：0db8：85a3：0：0：8a2e：0370：7334) 連線到負載平衡器。

考量事項

- 負載平衡器會根據目標群組的 IP 地址類型與目標進行通訊。
- 當您啟用負載平衡器的雙堆疊模式時，Elastic Load Balancing 會提供負載平衡器的 AAAA DNS 記錄。使用 IPv4 地址與負載平衡器通訊的用戶端可解析 A DNS 記錄。使用 IPv6 地址與負載平衡器通訊的用戶端可解析 AAAA DNS 記錄。
- 透過網際網路閘道存取內部雙堆疊負載平衡器會遭到封鎖，以防止來自網際網路的非預期存取。不過，這不會阻止非 IGW 網際網路存取（例如透過對等互連 AWS Direct Connect、Transit Gateway 或 Site-to-Site VPN)。
- Application Load Balancer 身分驗證僅在連線至 Identity Provider (IdP) 或 Amazon Cognito 端點時支援 IPv4。如果沒有公有 IPv4 地址，負載平衡器無法完成身分驗證程序，導致 HTTP 500 錯誤。

如需詳細資訊，請參閱更新 Application Load Balancer 的 IP 地址類型。

Application Load Balancer IP 地址管理

Application Load Balancer 使用來自 EC2 公有 IPv4 地址集區的公有彈性 IPv4 地址。 EC2 IPv4 使用 describe-addresses CLI、API 或在 AWS 主控台中檢視彈性 IPs(EIP) 區段 AWS 時，您的帳戶中可以看到這些 IP 地址。每個 ALB 關聯的 IP 地址都會標示為 service_managed 屬性設為 "ALB"。

雖然您的帳戶中可以看到這些 IPs，但它們仍由 Application Load Balancer 服務完全管理，且無法修改或釋出。Application Load Balancer 會在不再使用時，將 IPs 發行回公有 IPv4 地址集區。

CloudTrail 會記錄與 Application Load Balancer EIP 相關的 API 呼叫，例如「AllocateAddress」。服務委託人 'elasticloadbalancing.amazonaws.com' 會叫用這些 API 呼叫。

> **Note**
>
> 注意：Application Load Balancer 配置的 IPs 不會計入您帳戶的 EIP 限制。

IPAM IP 地址集區

IPAM IP 地址集區是您使用 Amazon VPC IP Address Manager (IPAM) 建立的連續 IP 地址範圍 （或 CIDRs) 集合。搭配 Application Load Balancer 使用 IPAM IP 地址集區可讓您根據路由和安全需求組織 IPv4 地址。IPAM IP 地址集區可讓您選擇將部分或全部公有 IPv4 地址範圍帶到 AWS ，並將其與 Application Load Balancer 搭配使用。啟動 EC2 執行個體和建立 Application Load Balancer 時，一律會優先使用您的 IPAM IP 地址集區。當您的 IP 地址不再使用時，會立即可供使用。

若要開始使用，請建立 IPAM IP 地址集區。如需詳細資訊，請參閱將 IP 地址帶入 IPAM。

考量事項

- 不支援 IPAM IPv6 地址集區。
- 內部負載平衡器或 IP 地址類型不支援 `dualstack-without-public-ipv4` IPAM IPv4 地址集區。
- 如果 IPAM IP 地址集區目前正由負載平衡器使用，則無法刪除 IP 地址。
- 在轉換到不同的 IPAM IP 地址集區期間，現有連線會根據負載平衡器 HTTP 用戶端保持連線持續時間終止。
- IPAM IP 地址集區可以跨多個帳戶共用。如需詳細資訊，請參閱設定 IPAM 的整合選項。
- 將 IPAM IP 地址集區與負載平衡器搭配使用不會產生額外費用。不過，視您使用的方案而定，IPAM 可能會產生相關費用。

如果您的 IPAM IP 地址集區中沒有更多可指派的 IP 地址，Elastic Load Balancing 會改用 AWS 受管 IPv4 地址。使用 AWS 受管 IPv4 地址需支付額外費用。若要避免這些成本，您可以將 IP 地址範圍新增至現有的 IPAM IP 地址集區。

如需詳細資訊，請參閱 Amazon VPC 定價。

負載平衡器連線

處理請求時，負載平衡器會維護兩個連線：一個與用戶端的連線，另一個與目標的連線。負載平衡器與用戶端之間的連線也稱為前端連線。負載平衡器與目標之間的連線也稱為後端連線。

跨區域負載平衡

使用 Application Load Balancer 時，跨區域負載平衡依預設會開啟，而且無法在負載平衡器層級進行變更。如需詳細資訊，請參閱 Elastic Load Balancing User Guide 中的 Cross-zone load balancing 章節。

可以在目標群組層級關閉跨區域負載平衡。如需詳細資訊，請參閱the section called “關閉跨區域負載平衡”。

DNS 名稱

每個 Application Load Balancer 都會收到具有下列語法的預設網域名稱系統 (DNS) 名稱：*name*-*id*.elb.*region*.amazonaws.com。例如，my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com。

如果您想要使用更易於記住的 DNS 名稱，您可以建立自訂網域名稱，並將其與 Application Load Balancer 的 DNS 名稱建立關聯。當用戶端使用此自訂網域名稱提出請求時，DNS 伺服器會解析為 Application Load Balancer 的 DNS 名稱。

首先，向取得認證的網域名稱註冊商註冊網域名稱。接著，使用您的 DNS 服務，例如網域註冊商，建立 DNS 記錄以將請求路由到您的 Application Load Balancer。如需詳細資訊，請參閱您的 DNS 服務文件。例如，如果您使用 Amazon Route 53 做為 DNS 服務，您可以建立指向 Application Load Balancer 的別名記錄。如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的將流量路由到 ELB 負載平衡器。

Application Load Balancer 每個已啟用的可用區域都有一個 IP 地址。這些是 Application Load Balancer 節點的 IP 地址。Application Load Balancer 的 DNS 名稱會解析為這些地址。例如，假設

Application Load Balancer 的自訂網域名稱為 `example.applicationloadbalancer.com`。使用下列 dig或 nslookup命令來判斷 Application Load Balancer 節點的 IP 地址。

Linux 或 Mac

```
$ dig +short example.applicationloadbalancer.com
```

Windows

```
C:\> nslookup example.applicationloadbalancer.com
```

Application Load Balancer 具有其節點的 DNS 記錄。您可以使用 DNS 名稱搭配下列語法來判斷 Application Load Balancer 節點的 IP 地址：*az*.*name*-*id*.elb.*region*.amazonaws.com。

Linux 或 Mac

```
$ dig +short us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

Windows

```
C:\> nslookup us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

建立 Application Load Balancer

Application Load Balancer 會從用戶端接收請求，並將其分佈到目標群組中的目標，例如 EC2 執行個體。如需詳細資訊，請參閱 Elastic Load Balancing 使用者指南中的 Elastic Load Balancing 的運作方式。 Elastic Load Balancing

任務

- 先決條件
- 建立負載平衡器
- 測試負載平衡器
- 後續步驟

先決條件

- 決定您的應用程式將支援哪些可用區域和 IP 地址類型。在每個可用區域中設定具有子網路的負載平衡器 VPC。如果應用程式同時支援 IPv4 和 IPv6 流量，請確保子網路同時具有 IPv4 和 IPv6 CIDRs。在每個可用區域中部署至少一個目標。如需詳細資訊，請參閱the section called “負載平衡器的子網路”。
- 確保目標執行個體的安全群組允許來自用戶端 IP 地址（如果目標由執行個體 ID 指定）或負載平衡器節點（如果目標由 IP 地址指定）的接聽程式連接埠流量。如需詳細資訊，請參閱建議的規則。
- 確保目標執行個體的安全群組使用運作狀態檢查通訊協定，允許來自運作狀態檢查連接埠上負載平衡器的流量。

建立負載平衡器

在建立 Application Load Balancer 的過程中，您將建立負載平衡器、至少一個接聽程式，以及至少一個目標群組。當每個啟用的可用區域中至少有一個運作狀態良好的註冊目標時，您的負載平衡器已準備好處理用戶端請求。

Console

建立 Application Load Balancer

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選擇 Create load balancer (建立負載平衡器)。
4. 在 Application Load Balancer (應用程式負載平衡器) 下，選擇 Create (建立)。
5. 基本組態
   a. 針對 Load balancer name (負載平衡器名稱)，輸入負載平衡器的名稱。名稱在 區域的一組負載平衡器中必須是唯一的。名稱最多可包含 32 個字元，而且只能包含英數字元和連字號。名稱開頭或結尾不得為連字號或 `internal-`。您無法在建立 Application Load Balancer 之後變更其名稱。
   b. 針對 Scheme (機制)，選擇 Internet-facing (面對網際網路) 或 internal (內部)。面對網際網路的負載平衡器會透過網際網路將用戶端的請求路由至目標。內部負載平衡器會使用私有 IP 地址將請求路由至目標。
   c. 針對負載平衡器 IP 地址類型，如果您的用戶端使用 IPvIPv4 地址與負載平衡器通訊，請選擇 IPv4；如果您的用戶端同時使用 IPv4 和 IPv6 地址與負載平衡器通訊，請選擇

Dualstack。 IPv4 如果您的用戶端僅使用 IPv6 地址與負載平衡器通訊，請選擇不含公有 IPv4 的雙堆疊。

6. 網路映射

   a. 針對 VPC，選取您為負載平衡器準備的 VPC。透過面向網際網路的負載平衡器，只有具有網際網路閘道VPCs 可供選取。

   b. （選用）對於 IP 集區，您可以選取將 IPAM 集區用於公有 IPv4 地址。如需詳細資訊，請參閱the section called “IPAM IP 地址集區”。

   c. 對於可用區域和子網路，為您的負載平衡器啟用區域，如下所示：

      - 從至少兩個可用區域選取子網路
      - 從至少一個本機區域選取子網路
      - 選擇一個 Outpost 子網路

      如需詳細資訊，請參閱the section called “負載平衡器的子網路”。

      使用 Dualstack 負載平衡器時，您必須選取同時具有 IPv4 和 IPv6 CIDR 區塊的子網路。

7. 安全群組

   我們會預先選取負載平衡器 VPC 的預設安全群組。您可以視需要選取其他安全群組。如果您沒有符合您需求的安全群組，請選擇建立新的安全群組以立即建立。如需詳細資訊，請參閱《Amazon VPC 使用者指南》的建立安全群組。

8. 接聽程式和路由

   a. 預設值為接聽程式，可接受連接埠 80 上的 HTTP 流量。您可保留預設接聽程式設定，或視需要修改通訊協定與連接埠。

   b. 針對預設動作，請選取要轉寄流量的目標群組。如果您沒有符合您需求的目標群組，請選擇建立目標群組以立即建立目標群組。如需詳細資訊，請參閱建立目標群組。

   c. （選用）選擇新增接聽程式標籤，然後輸入標籤索引鍵和標籤值。

   d. （選用）選擇新增接聽程式以新增另一個接聽程式（例如 HTTPS 接聽程式）。

9. 安全接聽程式設定

   只有在您新增 HTTPS 接聽程式時，才會顯示本節。

   a. 針對 Security policy (安全政策)，請選擇符合您需求的安全政策。如需詳細資訊，請參閱安全政策。

b. 針對預設 SSL/TLS 憑證，有下列選項可用：

- 如果您使用 建立或匯入憑證 AWS Certificate Manager，請選擇從 ACM，然後選擇憑證。
- 如果您使用 IAM 匯入憑證，請選擇從 IAM，然後選擇您的憑證。
- 如果您在 ACM 中沒有可用的憑證，但有可搭配負載平衡器使用的憑證，請選取匯入憑證並提供必要資訊。否則，請選擇請求新的 ACM 憑證。如需詳細資訊，請參閱AWS Certificate Manager 《使用者指南》中的AWS Certificate Manager 憑證。

c. （選用）選取相互驗證 (mTLS)，選擇政策以啟用 ALPN。

如需詳細資訊，請參閱交互 TLS 驗證。

10. 使用服務整合最佳化

（選用）您可以整合其他 AWS 與負載平衡器。如需詳細資訊，請參閱負載平衡器整合。

11. 負載平衡器標籤

（選用）展開負載平衡器標籤。選擇新增標籤，然後輸入標籤索引鍵和標籤值。如需詳細資訊，請參閱標籤。

12. 總結

複查您的組態，然後選擇 Create load balancer (建立負載平衡器)。建立期間會將一些預設屬性套用至 Network Load Balancer。您可以在建立 Network Load Balancer 之後檢視和編輯它們。如需詳細資訊，請參閱負載平衡器屬性。

AWS CLI

建立 Application Load Balancer

使用 create-load-balancer 命令。

下列範例會建立具有兩個已啟用可用區域和安全群組的面向網際網路負載平衡器。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type application \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

建立內部 Application Load Balancer

包含 `--scheme`選項，如下列範例所示。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type application \
    --scheme internal \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

建立雙堆疊 Application Load Balancer

包含 `--ip-address-type`選項，如下列範例所示。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type application \
    --ip-address-type dualstack \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

加入接聽程式

使用 create-listener 命令。如需範例，請參閱 建立 HTTP 接聽程式 和 建立 HTTPS 接聽程式。

CloudFormation

建立 Application Load Balancer

定義 AWS::ElasticLoadBalancingV2::LoadBalancer 類型的資源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

```
      Tags:
        - Key: "department"
          Value: "123"
```

加入接聽程式

定義 AWS::ElasticLoadBalancingV2::Listener 類型的資源。如需範例，請參閱 建立 HTTP 接聽程式 和 建立 HTTPS 接聽程式。

測試負載平衡器

建立負載平衡器後，請確認 EC2 執行個體已通過初始運作狀態檢查。然後，您可以檢查負載平衡器是否正在向 EC2 執行個體傳送流量。若要刪除負載平衡器，請參閱刪除 Application Load Balancer。

若要測試負載平衡器

1. 建立網路負載平衡器之後，選擇 Close (關閉)。
2. 在導覽窗格中，選擇 Target Groups (目標群組)。
3. 選取新建立的目標群組。
4. 選擇 Targets (目標) 並確認您的執行個體已就緒。如果執行個體的狀態為 `initial`，通常是因為執行個體仍在註冊中。此狀態也可能表示執行個體尚未通過最低數量的運作狀態檢查，無法視為運作狀態良好。至少有一個執行個體的運作狀態為健康之後，您可以測試您的負載平衡器。如需詳細資訊，請參閱目標運作狀態。
5. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
6. 選取新建立的負載平衡器。
7. 選擇描述，然後複製面向網際網路的負載平衡器或內部負載平衡器的 DNS 名稱 (例如 my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com)。

   - 對於面向網際網路的負載平衡器，將 DNS 名稱貼至已連接網際網路的 Web 瀏覽器的網址欄位。
   - 對於內部負載平衡器，請將 DNS 名稱貼至具有 VPC 私人連線的 Web 瀏覽器的網址欄位中。

   如果一切設定都正常，瀏覽器會顯示伺服器的預設頁面。
8. 如果網頁未顯示，請參閱下列文件以取得其他組態說明和疑難排解步驟。

   - 對於 DNS 相關問題，請參閱《Amazon Route 53 開發人員指南》中的將流量路由到 ELB 負載平衡器。

- 如需有關負載平衡器問題的資訊，請參閱為 Application Load Balancer 進行疑難排解。

後續步驟

建立負載平衡器之後，您可能想要執行下列動作：

- 新增接聽程式規則。
- 設定負載平衡器屬性。
- 設定目標群組屬性。
- 【HTTPS 接聽程式】 將憑證新增至選用憑證清單。
- 設定監控功能。

更新 Application Load Balancer 的可用區域

您可以隨時為您的負載平衡器啟用或停用可用區域。當您啟用可用區域之後，負載平衡器會開始將請求路由到該可用區域內已註冊的目標。Application Load Balancer 預設會在 上進行跨區域負載平衡，導致請求路由到所有可用區域的所有已註冊目標。當跨區域負載平衡關閉時，負載平衡器只會將請求路由到相同可用區域中的目標。如需詳細資訊，請參閱跨區域負載平衡。如果您確認每個已啟用的可用區域擁有至少一個登錄的目標，您的負載平衡器會展現最高效率。

當您停用可用區域之後，該可用區域內的目標仍註冊到負載平衡器，但負載平衡器不會將請求路由給它們。

如需詳細資訊，請參閱the section called “負載平衡器的子網路”。

Console

更新可用區域

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在網路映射索引標籤中，選擇編輯子網路。
5. 若要啟用可用區域，請選取其核取方塊並選取子網路。如果可用的子網路只有一個，則會選取該子網路。

6. 若要變更已啟用可用區域的子網路，請從清單中選擇其中一個其他的子網路。
7. 若要停用可用區域，請清除其核取方塊。
8. 選擇儲存變更。

AWS CLI

更新可用區域

使用 set-subnets 命令。

```
aws elbv2 set-subnets \
    --load-balancer-arn load-balancer-arn \
    --subnets subnet-8360a9e7EXAMPLE subnet-b7d581c0EXAMPLE
```

CloudFormation

更新可用區域

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref new-subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

Application Load Balancer 的安全群組

Application Load Balancer 的安全群組會控制允許到達和離開負載平衡器的流量。您必須確保負載平衡器可以在接聽程式連接埠和運作狀態檢查連接埠上與已註冊的目標通訊。當您將接聽程式新增到負載平

衡器，或針對目標群組更新負載平衡器用來路由請求的運作狀態檢查連接埠時，您必須確認與負載平衡器相關聯的安全群組在新的連接埠上允許這兩個方向的流量。如果不是如此，您可以編輯目前相關聯之安全群組的規則，或將其他安全群組與負載平衡器建立關聯。您可以選擇要允許的連接埠和通訊協定。例如，您可以開放網際網路控制訊息通訊協定 (ICMP) 連線負載平衡器回應 ping 請求 (不過，ping 請求不會轉發到任何執行個體)。

考量事項

- 為了確保您的目標僅接收來自負載平衡器的流量，請將與目標相關聯的安全群組限制為僅接受來自負載平衡器的流量。這可以透過將負載平衡器的安全群組設定為目標安全群組的輸入規則中的來源來實現。
- 如果您的 Application Load Balancer 是 Network Load Balancer 的目標，Application Load Balancer 的安全群組會使用連線追蹤來追蹤來自 Network Load Balancer 的流量相關資訊。無論為 Application Load Balancer 設定的安全群組規則為何，都會發生此情況。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的安全群組連線追蹤。
- 建議您允許傳入 ICMP 流量以支援路徑 MTU 探索。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的路徑 MTU 探索。

建議的規則

對於執行個體做為目標的面向網際網路負載平衡器，建議使用下列規則。

Inbound

來源	連接埠範圍	Comment
0.0.0.0/0	####	在負載平衡器接聽程式連接埠上允許所有傳入流量

Outbound

目標	連接埠範圍	Comment
########	########	在執行個體接聽程式連接埠上允許流向執行個體的傳出流量
########	######	在運作狀態檢查連接埠上允許流向執行個體的傳出流量

對於執行個體做為目標的內部負載平衡器，建議使用下列規則。

Inbound

來源	連接埠範圍	Comment
VPC CIDR	*####*	在負載平衡器接聽程式連接埠上允許來自 VPC CIDR 的傳入流量

Outbound

目標	連接埠範圍	Comment
########	*########*	在執行個體接聽程式連接埠上允許流向執行個體的傳出流量
########	*######*	在運作狀態檢查連接埠上允許流向執行個體的傳出流量

對於執行個體本身為 Network Application Load Balancer 目標的 Application Load Balancer，建議使用下列規則。

Inbound

來源	連接埠範圍	Comment
### IP ##/CIDR	*alb ####*	允許負載平衡器接聽程式連接埠上的傳入用戶端流量
VPC CIDR	*alb ####*	允許透過負載平衡器接聽程式連接埠 AWS PrivateLink 上的傳入用戶端流量
VPC CIDR	*alb ####*	允許來自 Network Load Balancer 的傳入運作狀態檢查流量

Outbound

目標	連接埠範圍	Comment
########	########	在執行個體接聽程式連接埠上允許流向執行個體的傳出流量
########	######	在運作狀態檢查連接埠上允許流向執行個體的傳出流量

更新相關聯的安全群組

您可以隨時更新與負載平衡器相關聯的安全群組。

Console

更新安全群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在安全性索引標籤中，選擇編輯。
5. 若要將安全群組與負載平衡器建立關聯，請選取安全群組。若要移除安全群組關聯，請選擇安全群組的 X 圖示。
6. 選擇儲存變更。

AWS CLI

更新安全群組

使用 set-security-groups 命令。

```
aws elbv2 set-security-groups \
    --load-balancer-arn load-balancer-arn \
    --security-groups sg-01dd3383691d02f42 sg-00f4e409629f1a42d
```

CloudFormation

更新安全群組

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
        - !Ref myNewSecurityGroup
```

更新 Application Load Balancer 的 IP 地址類型

您可以設定 Application Load Balancer，讓用戶端只能使用 IPv4 地址，或既可使用 IPv4 又可使用 IPv6 地址 (雙堆疊)，來與負載平衡器通訊。負載平衡器會根據目標群組的 IP 地址類型與目標進行通訊。如需詳細資訊，請參閱IP 地址類型。

雙堆疊要求

- 您可以在建立負載平衡器時設定 IP 地址類型，並隨時更新它。
- 您為負載平衡器指定的 Virtual Private Cloud (VPC) 和子網路必須具有相關聯的 IPv6 CIDR 區塊。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的 IPv6 地址。
- 負載平衡器子網路的路由表必須路由 IPv6 流量。
- 負載平衡器的安全群組必須允許 IPv6 流量。
- 負載平衡器子網路的網路 ACL 必須允許 IPv6 流量。

Console

更新 IP 地址類型

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。

4. 在網路映射索引標籤上，選擇編輯 IP 地址類型。
5. 對於 IP 地址類型，選擇 IPv4 僅支援 IPv4 地址、選擇 Dualstack 同時支援 IPv4 和 IPv6 地址，或選擇不使用公有 IPv4 的 Dualstack 僅支援 IPv6 地址。
6. 選擇儲存變更。

AWS CLI

更新 IP 地址類型

使用 set-ip-address-type 命令。

```
aws elbv2 set-ip-address-type \
    --load-balancer-arn load-balancer-arn \
    --ip-address-type dualstack
```

CloudFormation

更新 IP 地址類型

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

更新 Application Load Balancer 的 IPAM IP 地址集區

IPAM IP 地址集區必須先在 IPAM 內建立，然後 Application Load Balancer 才能使用。如需詳細資訊，請參閱將 IP 地址帶入 IPAM。

Console

更新 IPAM IP 地址集區

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在網路映射索引標籤上，選擇編輯 IP 集區。
5. 在 IP 集區下，選取將 IPAM 集區用於公有 IPv4 地址，然後選擇 IPAM 集區。
6. 選擇儲存變更。

AWS CLI

更新 IPAM IP 地址集區

使用 modify-ip-pools 命令。

```
aws elbv2 modify-ip-pools \
    --load-balancer-arn load-balancer-arn \
    --ipam-pools Ipv4IpamPoolId=ipam-pool-1234567890abcdef0
```

CloudFormation

更新 IPAM IP 地址集區

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internet-facing
      IpAddressType: ipv4
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

```
      Ipv4IpamPoolId: !Ref myIPAMPool
```

編輯 Application Load Balancer 的屬性

建立 Application Load Balancer 之後，您可以編輯其屬性。

負載平衡器屬性

- 連線閒置逾時
- HTTP 用戶端保持連線持續時間
- 刪除保護
- 去同步緩解模式
- 主機標頭保留

連線閒置逾時

連線閒置逾時是指在負載平衡器關閉連線之前，現有用戶端或目標連線可以保持非作用中狀態且未傳送或接收資料的時間。

為了確保檔案上傳等冗長操作有時間完成，請在每個閒置逾時期間經過之前傳送至少 1 個位元組的資料，並視需要增加閒置逾時期間的長度。也建議您將應用程式的閒置逾時設定為大於負載平衡器所設定的閒置逾時。否則，如果應用程式不正常地關閉與負載平衡器的 TCP 連線，負載平衡器可能會在收到指示連線已關閉的封包之前傳送請求給應用程式。如果是這種情況，則負載平衡器會向用戶端傳送「HTTP 502 錯誤閘道」錯誤。

Application Load Balancer 不支援 HTTP/2 PING 影格。這些不會重設連線閒置逾時。

預設情況下，Elastic Load Balancing 會將負載平衡器的閒置逾時設為 60 秒。

Console

更新連線閒置逾時值

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在屬性索引標籤中，選擇編輯。
5. 在流量組態下，輸入連線閒置逾時的值。有效範圍為 1 到 4000 秒。

6. 選擇儲存變更。

AWS CLI

更新連線閒置逾時值

以 屬性來使用 modify-load-balancer-attributes命令。有效範圍為 1 到 4000 秒。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=idle_timeout.timeout_seconds,Value=120"
```

CloudFormation

更新連線閒置逾時值

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 idle_timeout.timeout_seconds 屬性。有效範圍為 1 到 4000 秒。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "idle_timeout.timeout_seconds"
          Value: "120"
```

HTTP 用戶端保持連線持續時間

HTTP 用戶端保持連線持續時間是 Application Load Balancer 持續維護用戶端 HTTP 連線的時間長度上限。經過設定的 HTTP 用戶端保持連線持續時間後，Application Load Balancer 會接受另一個請求，然後傳回可正常關閉連線的回應。

負載平衡器傳送的回應類型取決於用戶端連線所使用的 HTTP 版本。

- 對於使用 HTTP 1.x 連線的用戶端，負載平衡器會傳送包含 欄位 的 HTTP 標頭。 `Connection: close`
- 對於使用 HTTP/2 連線的用戶端，負載平衡器會傳送GOAWAY影格。

根據預設，Application Load Balancer 會將負載平衡器的 HTTP 用戶端持續作用持續時間值設定為 3600 秒或 1 小時。HTTP 用戶端保持連線持續時間無法關閉或設定為低於最短 60 秒，但您可以增加 HTTP 用戶端保持連線持續時間，最長可達 604800 秒，或 7 天。Application Load Balancer 會在最初建立與用戶端的 HTTP 連線時，開始 HTTP 用戶端保持連線持續時間。當沒有流量時，持續時間會繼續，且在建立新連線之前不會重設。

當負載平衡器流量使用區域轉移或區域自動轉移從受損的可用區域轉移時，具有現有開放連線的用戶端可能會繼續對受損位置提出請求，直到用戶端重新連線為止。若要支援更快的復原，請考慮設定較低的持續作用持續時間值，以限制用戶端與負載平衡器保持連線的時間量。如需詳細資訊，請參閱《Amazon Application Recovery Controller (ARC) 開發人員指南》中的限制用戶端與您的端點保持連線的時間。

> **Note**
>
> 當負載平衡器將 Application Load Balancer 的 IP 地址類型切換至 時`dualstack-without-public-ipv4`，負載平衡器會等待所有作用中連線完成。若要減少切換 Application Load Balancer IP 地址類型所需的時間，請考慮降低 HTTP 用戶端保持連線持續時間。

Application Load Balancer 會在初始連線期間指派 HTTP 用戶端保持連線持續時間值。當您更新 HTTP 用戶端保持連線持續時間時，這可能會導致與不同的 HTTP 用戶端保持連線持續時間值同時連線。現有的連線會保留初始連線期間套用的 HTTP 用戶端持續作用持續時間值。新的連線會收到更新的 HTTP 用戶端保持連線持續時間值。

Console

更新用戶端持續作用持續時間

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在屬性索引標籤中，選擇編輯。

5. 在流量組態下，輸入 HTTP 用戶端保持連線持續時間的值。有效範圍為 60 到 604800 秒。
6. 選擇儲存變更。

AWS CLI

更新用戶端持續作用持續時間

以 屬性來使用 modify-load-balancer-attributesclient_keep_alive.seconds 命令。有效範圍為 60 到 604800 秒。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=client_keep_alive.seconds,Value=7200"
```

CloudFormation

更新用戶端持續作用持續時間

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 client_keep_alive.seconds 屬性。有效範圍為 60 到 604800 秒。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "client_keep_alive.seconds"
          Value: "7200"
```

刪除保護

為避免您的負載平衡器上遭意外刪除，您可以啟用刪除保護。您的負載平衡器的刪除保護預設為停用。

如果您為負載平衡器啟用刪除保護，則必須先停用才可刪除負載平衡器。

Console

啟用或停用刪除保護

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在屬性索引標籤中，選擇編輯。
5. 在保護下，啟用或停用刪除保護。
6. 選擇儲存變更。

AWS CLI

啟用或停用刪除保護

以 屬性來使用 modify-load-balancer-attributes`deletion_protection.enabled` 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=deletion_protection.enabled,Value=true"
```

CloudFormation

啟用或停用刪除保護

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 `deletion_protection.enabled` 屬性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
```

```
      - !Ref subnet-AZ2
    SecurityGroups:
      - !Ref mySecurityGroup
    LoadBalancerAttributes:
      - Key: "deletion_protection.enabled"
        Value: "true"
```

去同步緩解模式

去同步緩解模式可保護應用程式免於因 HTTP 去同步而發生問題。負載平衡器會根據其威脅層級對每個要求進行分類，允許安全要求，然後根據您指定的緩和模式來降低風險。非同步緩和模式分為監控、防禦性和最嚴格。預設值為防禦模式，可針對 HTTP 非同步提供持久的緩和措施，同時維持應用程式的可用性。您可以切換至最嚴格模式，以確保應用程式只接收符合 RFC 7230 的請求。

http_desync_guardian 程式庫會分析 HTTP 請求，以防止 HTTP 去同步攻擊。如需詳細資訊，請參閱 GitHub 上的 HTTP Desync Guardian。

分類

分類如下：

- 合規 — 要求符合 RFC 7230，不會造成任何已知的安全威脅。
- 可接受 — 要求不符合 RFC 7230，但不會造成已知的安全威脅。
- 不明確 — 要求不符合 RFC 7230，但造成風險，因為各種 Web 伺服器和代理的處理方式不同。
- 嚴重 — 要求造成高安全性風險。負載平衡器會封鎖要求，傳送提供 400 回應至用戶端，並關閉用戶端連線。

如果要求不符合 RFC 7230，負載平衡器會增加 `DesyncMitigationMode_NonCompliant_Request_Count` 指標。如需詳細資訊，請參閱Application Load Balancer 指標。

每個請求的分類都包含在負載平衡器存取日誌中。如果請求不符合規定，存取日誌會包含分類原因代碼。如需詳細資訊，請參閱分類原因。

模式

下表說明 Application Load Balancer 如何根據模式和分類處理請求。

分類	監控模式	防禦性模式	最嚴格模式
合規	允許	已允許	允許
可接受	允許	允許	封鎖
不明確	允許	允許[1]	封鎖
嚴重	允許	封鎖	封鎖

[1] 路由傳送要求，但關閉用戶端和目標連接。如果負載平衡器在防禦模式下收到大量「不明確」請求，則可能會產生額外費用。這是因為每秒增加的新連線數目會提高每小時使用的負載平衡器容量單位 (LCU)。您可以使用 NewConnectionCount 指標，來比較負載平衡器在監控模式和防禦模式下建立新連線的方式。

Console

更新非同步緩解模式

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在屬性索引標籤中，選擇編輯。
5. 在流量組態、封包處理下，針對解除同步緩解模式，選擇防禦、嚴格或監控。
6. 選擇儲存變更。

AWS CLI

更新非同步緩解模式

以 屬性來使用 modify-load-balancer-attributesrouting.http.desync_mitigation_mode 命令。可能的值為 monitor、defensive 或 strictest。預設值為 defensive。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=routing.http.desync_mitigation_mode,Value=monitor"
```

CloudFormation

更新非同步緩解模式

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 routing.http.desync_mitigation_mode 屬性。可能的值為 monitor、defensive 或 strictest。預設值為 defensive。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "routing.http.desync_mitigation_mode"
          Value: "monitor"
```

主機標頭保留

啟用保留主機標頭屬性後，Application Load Balancer 會保留 HTTP 請求中的 Host 標頭，並將標頭傳送到目標，而不進行任何修改。如果 Application Load Balancer 收到多個 Host 標頭，則會全部保留。只會將接聽程式規則套用至收到的第一個 Host 標頭。

依預設，如果未啟用保留主機標頭屬性，則 Application Load Balancer 會以下列方式修改 Host 標頭：

未啟用主機標頭保留，且接聽程式連接埠不是預設連接埠時：未使用預設連接埠 (連接埠 80 或 443) 時，如果用戶端尚未附加連接埠號碼，則我們會將該號碼附加至主機標頭。例如，如果接聽程式連接埠不是預設連接埠 (例如 8080)，則內含 Host: www.example.com 之 HTTP 請求中的 Host 標頭會修改為 Host: www.example.com:8080。

未啟用主機標頭保留，且接聽程式連接埠為預設連接埠 (連接埠 80 或 443) 時：對於預設的接聽程式連接埠 (連接埠 80 或 443)，我們不會將連接埠號碼附加至傳出主機標頭。已存在於傳入主機標頭中的任何連接埠號碼都會遭到移除。

下表顯示更多範例，說明 Application Load Balancer 如何根據接聽程式連接埠處理 HTTP 請求中的主機標頭。

接聽程式連接埠	範例請求	請求中的主機標頭	主機標頭保留已停用 (預設行為)	主機標頭保留已啟用
在預設的 HTTP/HTTPS 接聽程式上傳送請求。	`GET / index.html HTTP/1.1 Host: example.com`	example.com	example.com	example.com
請求會在預設 HTTP 接聽程式上傳送，而主機標頭具有連接埠（例如 80 或 443)。	`GET / index.html HTTP/1.1 Host: example.com:80`	example.com:80	example.com	example.com:80
請求具有絕對路徑。	`GET https://dns_name/index.html HTTP/1.1 Host: example.com`	example.com	dns_name	example.com
請求會在非預設接聽程式連接埠（例如 8080) 上傳送	`GET / index.html HTTP/1.1 Host: example.com`	example.com	example.com:8080	example.com
在非預設接聽程式連接埠上傳送請求，並且主機標頭具有連接埠 (例如 8080)。	`GET / index.html HTTP/1.1 Host:`	example.com:8080	example.com:8080	example.com:8080

接聽程式連接埠	範例請求	請求中的主機標頭	主機標頭保留已停用 (預設行為)	主機標頭保留已啟用
	`example.com:8080`			

Console

啟用主機標頭保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在屬性索引標籤中，選擇編輯。
5. 在封包處理下方，開啟保留主機標頭。
6. 選擇儲存變更。

AWS CLI

啟用主機標頭保留

使用 modify-load-balancer-attributes 命令，同時將 `routing.http.preserve_host_header.enabled` 屬性設為 `true`。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=routing.http.preserve_host_header.enabled,Value=true"
```

CloudFormation

啟用主機標頭保留

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 `routing.http.preserve_host_header.enabled` 屬性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
```

```
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "routing.http.preserve_host_header.enabled"
          Value: "true"
```

標記 Application Load Balancer

標籤可幫助您以不同的方式來將負載平衡器分類，例如，根據目的、擁有者或環境。

您可以在每個負載平衡器中加入多個標籤。如果所新增的標籤，其索引鍵已經與負載平衡器相關聯，則此動作會更新該標籤的值。

使用標籤完成負載平衡器使用後，可將其自負載平衡器中移除。

限制

- 每一資源標籤數上限：50
- 索引鍵長度上限：127 個 Unicode 字元
- 數值長度上限：255 個 Unicode 字元
- 標籤鍵與值皆區分大小寫。允許的字元包括可用 UTF-8 表示的英文字母、空格和數字，還有以下特殊字元：+ - = . _ : / @。不可使用結尾或前方空格。
- 請勿在標籤名稱或值中使用 `aws:`字首，因為它保留供 AWS 使用。您不可編輯或刪除具此字首的標籤名稱或值。具此字首的標籤，不算在受資源限制的標籤計數內。

Console

更新負載平衡器的標籤

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。

3. 選取負載平衡器。
4. 在 Tags (標籤) 索引標籤上，選擇 Manage tags (管理標籤)。
5. 若要新增標籤，請選取新增標籤，然後輸入標籤金鑰與值。
6. 若要更新標籤，請在金鑰或值中輸入新值。
7. 若要刪除標籤，請選擇標籤旁的移除。
8. 選擇儲存變更。

AWS CLI

新增 標籤

使用 add-tags 命令。

```
aws elbv2 add-tags \
    --resource-arns load-balancer-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

移除標籤

使用 remove-tags 命令。

```
aws elbv2 remove-tags \
    --resource-arns load-balancer-arn \
    --tag-keys project department
```

CloudFormation

新增 標籤

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 Tags 屬性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
```

```
      - !Ref subnet-AZ2
    SecurityGroups:
      - !Ref mySecurityGroup
    Tags:
      - Key: 'project'
        Value: 'lima'
      - Key: 'department'
        Value: 'digital-media'
```

刪除 Application Load Balancer

在您的負載平衡器可用後，將會根據持續執行時間收取一小時或不足一小時的費用。當您不再需要負載平衡器時，可以將它刪除。刪除負載平衡器後，便會停止收取費用。

如果已啟用刪除保護，則無法刪除負載平衡器。如需詳細資訊，請參閱刪除保護。

請注意，刪除負載平衡器不會影響其登錄目標。例如，您的 EC2 執行個體將繼續執行，且仍會登錄到他們的目標群組。若要刪除您的目標群組，請參閱刪除 Application Load Balancer 目標群組。

DNS 記錄

如果您網域有指向負載平衡器的 DNS 記錄，請將其指向新的位置，並等待 DNS 變更生效，然後刪除負載平衡器。

- 如果記錄是存留時間 (TTL) 為 300 秒的 CNAME 記錄，請等待至少 300 秒，然後再繼續執行下一個步驟。
- 如果記錄是 Route 53 別名 (A) 記錄，請至少等待 60 秒。
- 如果使用 Route 53，則記錄變更需要 60 秒才能傳播到所有全域 Route 53 名稱伺服器。將此時間新增至正在更新之記錄的 TTL 值。

Console

刪除負載平衡器

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器，然後選擇動作、刪除負載平衡器。
4. 出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

AWS CLI

刪除負載平衡器

使用 delete-load-balancer 指令。

```
aws elbv2 delete-load-balancer \
    --load-balancer-arn load-balancer-arn
```

檢視 Application Load Balancer 資源映射

Application Load Balancer 資源映射提供負載平衡器架構的互動式顯示，包括其關聯的接聽程式、規則、目標群組和目標。資源映射也會反白顯示所有資源之間的關係和路由路徑，產生負載平衡器組態的視覺化呈現。

檢視 Application Load Balancer 的資源映射

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 選擇資源映射索引標籤，以顯示負載平衡器的資源映射。

資源地圖元件

地圖檢視

Application Load Balancer 資源映射中有兩個可用檢視：概觀和運作狀態不佳的目標映射。預設會選取概觀，並顯示所有負載平衡器的資源。選取運作狀態不佳的目標映射檢視只會顯示運作狀態不佳的目標和與其相關聯的資源。

運作狀態不佳的目標映射檢視可用來疑難排解運作狀態檢查失敗的目標。如需詳細資訊，請參閱使用資源映射對運作狀態不佳的目標進行故障診斷。

Resource Groups

Application Load Balancer 資源映射包含四個資源群組，每個資源類型各一個。資源群組是接聽程式、規則、目標群組和目標。

資源圖磚

群組中的每個資源都有自己的圖磚，顯示該特定資源的詳細資訊。

- 將滑鼠暫留在資源圖磚上會反白顯示資源與其他資源之間的關係。
- 選取資源圖磚會反白顯示資源與其他資源之間的關係，並顯示該資源的其他詳細資訊。
  - 規則條件：每個規則的條件。
  - 目標群組運作狀態摘要：每個運作狀態的已註冊目標數量。
  - 目標運作狀態 目標為目前的運作狀態和描述。

> **Note**
>
> 您可以關閉顯示資源詳細資訊以在資源映射中隱藏其他詳細資訊。

- 每個資源圖磚都包含一個連結，當選取時，該連結會導覽至該資源的詳細資訊頁面。
  - 接聽程式 - 選取接聽程式通訊協定：連接埠。例如 `HTTP:80`
  - 規則 - 選取規則動作。例如 `Forward to target group`
  - 目標群組 - 選取目標群組名稱。例如 `my-target-group`
  - 目標 - 選取目標 ID。例如 `i-1234567890abcdef0`

匯出資源映射

選取匯出可讓您選擇將 Application Load Balancer 資源映射的目前檢視匯出為 PDF。

Application Load Balancer 的區域轉移

區域轉移和區域自動轉移是 Amazon Application Recovery Controller (ARC) 的功能。透過區域轉移，您可以透過單一動作將流量從受損的可用區域轉移。如此一來，您就可以繼續從 AWS 區域中其他運作狀況良好的可用區域進行操作。

使用區域自動轉移時，您授權 AWS 代表您在事件期間從可用區域轉移應用程式的資源流量，以協助縮短復原時間。當內部監控指出有可能影響客戶的可用區域受損時， 會 AWS 啟動自動轉移。當 AWS 啟動自動轉移時，您為區域自動轉移設定之資源的應用程式流量會開始從可用區域轉移。

當您開始區域轉移時，負載平衡器會停止將資源的新流量傳送至受影響的可用區域。ARC 會立即建立區域轉移。不過，根據用戶端行為和連線重複使用，在可用區域中現有的進行中連線可能需要一小段時間才能完成。根據您的 DNS 設定和其他因素，現有的連線可能會在幾分鐘內完成，或者可能需要更長

的時間。如需詳細資訊，請參閱《Amazon Application Recovery Controller (ARC) 開發人員指南》中的限制用戶端與您的端點保持連線的時間。

目錄

- 開始區域轉移之前
- 跨區域負載平衡
- 區域轉移管理覆寫
- 為您的 Application Load Balancer 啟用區域轉移
- 為您的 Application Load Balancer 啟動區域轉移
- 更新 Application Load Balancer 的區域轉移
- 取消 Application Load Balancer 的區域轉移

開始區域轉移之前

- 區域轉移預設為停用，且必須在每個 Application Load Balancer 上啟用。如需詳細資訊，請參閱為您的 Application Load Balancer 啟用區域轉移。
- 您只能針對單一可用區域，啟動特定負載平衡器的區域轉移。您無法為多個可用區域啟動區域轉移。
- AWS 當多個基礎設施問題影響 服務時， 會主動從 DNS 移除區域負載平衡器 IP 地址。在啟動區域轉移之前，請務必檢查目前的可用區域容量。如果您的負載平衡器已關閉跨區域負載平衡，而您使用區域轉移來移除區域負載平衡器 IP 地址，則受區域轉移影響的可用區域也會失去目標容量。

如需詳細資訊，請參閱《Amazon Application Recovery Controller (ARC) 開發人員指南》中的 ARC 中的區域轉移最佳實務。

跨區域負載平衡

在啟用跨區域負載平衡的 Application Load Balancer 上啟動區域轉移時，對目標的所有流量都會在受影響的可用區域中遭到封鎖，並從 DNS 中移除區域 IP 地址。

優點：

- 從可用區域故障中更快速復原。
- 在可用區域中偵測到故障時，能夠將流量移至運作狀態良好的可用區域。
- 您可以透過模擬和識別故障來測試應用程式完整性，以防止意外停機時間。

區域轉移管理覆寫

屬於 Application Load Balancer 的目標包含獨立`AdministrativeOverride`於 `TargetHealth` 狀態的新狀態 。

為 Application Load Balancer 啟動區域轉移時，區域內所有要移離的目標都會視為在管理上遭到覆寫。Application Load Balancer 會停止將新流量路由到管理上覆寫的目標。現有的連線會保持不變，直到它們以有機方式關閉為止。

可能`AdministrativeOverride`的狀態為：

不明

由於內部錯誤，無法傳播狀態

no_override

目標上目前沒有作用中的覆寫

zonal_shift_active

區域轉移在目標可用區域中處於作用中狀態

為您的 Application Load Balancer 啟用區域轉移

區域轉移預設為停用，且必須在每個 Application Load Balancer 上啟用。這可確保您只能使用您想要的特定 Application Load Balancer 啟動區域轉移。如需詳細資訊，請參閱the section called “區域轉移”。

Console

啟用區域轉移

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡器)，選擇 Load Balancer (負載平衡器)。
3. 選取 Application Load Balancer。
4. 在屬性索引標籤中，選擇編輯。
5. 在可用區域路由組態下，針對 ARC 區域轉移整合，選擇啟用。
6. 選擇儲存變更。

AWS CLI

啟用區域轉移

以 屬性來使用 modify-load-balancer-attributeszonal_shift.config.enabled 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=zonal_shift.config.enabled,Value=true"
```

CloudFormation

啟用區域轉移

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 zonal_shift.config.enabled 屬性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        -Key: "zonal_shift.config.enabled"
         Value: "true"
```

為您的 Application Load Balancer 啟動區域轉移

ARC 中的區域轉移可讓您暫時將支援資源的流量移離可用區域，讓您的應用程式可以繼續與 AWS 區域中的其他可用區域正常運作。

先決條件

開始之前，請確認您已啟用負載平衡器的區域轉移。

Console

此程序說明如何使用 Amazon EC2 主控台啟動區域轉移。如需使用 ARC 主控台啟動區域轉移的步驟，請參閱《Amazon Application Recovery Controller (ARC) 開發人員指南》中的啟動區域轉移。

啟動區域轉移

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡器)，選擇 Load Balancer (負載平衡器)。
3. 選取 Application Load Balancer。
4. 在整合索引標籤上，展開 Amazon Application Recovery Controller (ARC)，然後選擇開始區域轉移。
5. 選取要將流量移出的可用區域。
6. 選擇或輸入區域轉移的到期日。區域轉移最初可設定為 1 分鐘至三天 (72 小時)。

   所有區域轉移都是暫時的。您必須設定到期日，但您可以稍後更新作用中的轉移以設定新的到期日。
7. 輸入註解。您可以稍後更新區域轉移來編輯註解。
8. 選取核取方塊，確認啟動區域轉移可透過將流量移離可用區域來減少應用程式的容量。
9. 選擇確認。

AWS CLI

啟動區域轉移

使用 Amazon Application Recovery Controller (ARC) start-zonal-shift 命令。

```
aws arc-zonal-shift start-zonal-shift \
    --resource-identifier load-balancer-arn \
    --away-from use2-az2 \
    --expires-in 2h \
    --comment "zonal shift due to scheduled maintenance"
```

更新 Application Load Balancer 的區域轉移

您可以更新區域轉移以設定新的過期，或編輯或取代區域轉移的註解。

Console

此程序說明如何使用 Amazon EC2 主控台更新區域轉移。如需使用 Amazon Application Recovery Controller (ARC) 主控台更新區域轉移的步驟，請參閱《Amazon Application Recovery Controller (ARC) 開發人員指南》中的更新區域轉移。

更新區域轉移

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡器)，選擇 Load Balancer (負載平衡器)。
3. 選取具有作用中區域轉移的 Application Load Balancer。
4. 在整合索引標籤上，展開 Amazon Application Recovery Controller (ARC)，然後選擇更新區域轉移。

   這會開啟 ARC 主控台以繼續更新程序。
5. （選用）對於設定區域轉移過期，選取或輸入過期。
6. （選用）對於註解，選擇性編輯現有的註解或輸入新的註解。
7. 選擇更新。

AWS CLI

更新區域轉移

使用 Amazon Application Recovery Controller (ARC) update-zonal-shift 命令。

```
aws arc-zonal-shift update-zonal-shift \
    --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE \
    --expires-in 1h \
    --comment "extending zonal shift for scheduled maintenance"
```

取消 Application Load Balancer 的區域轉移

您可以在區域轉移過期之前隨時取消區域轉移。您可以取消您啟動的區域轉移，或為區域自動轉移實務執行的資源 AWS 啟動的區域轉移。

Console

此程序說明如何使用 Amazon EC2 主控台取消區域轉移。如需使用 Amazon Application Recovery Controller (ARC) 主控台取消區域轉移的步驟，請參閱《Amazon Application Recovery Controller (ARC) 開發人員指南》中的取消區域轉移。

取消區域轉移

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡器)，選擇 Load Balancer (負載平衡器)。
3. 選取具有作用中區域轉移的 Application Load Balancer。
4. 在整合索引標籤的 Amazon Application Recovery Controller (ARC) 下，選擇取消區域轉移。

   這會開啟 ARC 主控台以繼續取消程序。
5. 選擇取消區域轉移。
6. 出現確認提示時，選擇 Confirm (確認)。

AWS CLI

取消區域轉移

使用 Amazon Application Recovery Controller (ARC) cancel-zonal-shift 命令。

```
aws arc-zonal-shift cancel-zonal-shift \
    --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE
```

Application Load Balancer 的容量保留

負載平衡器容量單位 (LCU) 保留可讓您為負載平衡器保留靜態最小容量。Application Load Balancer 會自動擴展以支援偵測到的工作負載，並滿足容量需求。設定最小容量時，您的負載平衡器會根據收到的流量繼續向上或向下擴展，但也會防止容量低於設定的最小容量。

考慮在下列情況下使用 LCU 保留：

- 您即將發生的事件會有突然、不尋常的高流量，並希望確保您的負載平衡器能夠在事件期間支援突增的流量尖峰。
- 由於工作負載的性質，您在短時間內會有無法預測的尖峰流量。

- 您要將負載平衡器設定為在特定開始時間加入或遷移服務，並且需要從高容量開始，而不是等待自動擴展生效。
- 您正在負載平衡器之間遷移工作負載，並想要設定目的地以符合來源的規模。

估算您需要的容量

判斷負載平衡器應預留的容量時，建議您執行負載測試或檢閱代表您預期近期流量的歷史工作負載資料。使用 Elastic Load Balancing 主控台，您可以根據檢閱的流量預估需要預留多少容量。

或者，您可以使用 CloudWatch 指標PeakLCUs來判斷所需的容量層級。PeakLCUs 指標會考慮流量模式中的峰值，負載平衡器必須跨所有擴展維度進行擴展，以支援工作負載。PeakLCUs 指標與 ConsumedLCUs 指標不同，該指標只會彙總流量的帳單維度。建議使用 PeakLCUs 指標，以確保您的 LCU 保留在負載平衡器擴展期間是足夠的。估計容量時，請使用每分鐘 Sum PeakLCUs。

如果您沒有歷史工作負載資料可供參考，且無法執行負載測試，您可以使用 LCU 保留計算器預估所需的容量。LCU 保留計算器會根據 AWS 觀察到的歷史工作負載使用資料，可能不會代表您的特定工作負載。如需詳細資訊，請參閱Load Balancer容量單位保留計算器。

LCU 保留的最小值和最大值

保留請求總數必須至少為 100 LCU。最大值取決於您帳戶的配額。如需詳細資訊，請參閱the section called “Load Balancer容量單位”。

請求 Application Load Balancer 的負載平衡器容量單位保留

在使用 LCU 保留之前，請檢閱下列項目：

- 容量會保留在區域層級，並平均分散到各個可用區域。在開啟 LCU 保留之前，請確認每個可用區域中有足夠的平均分佈目標。
- LCU 保留請求是以先到先得的方式完成，並且取決於當時區域的可用容量。大多數請求通常在幾分鐘內完成，但最多可能需要幾個小時。
- 若要更新現有的保留，先前的請求必須佈建或失敗。您可以視需要增加預留容量，但每天只能減少兩次預留容量。
- 您將繼續支付任何預留或佈建容量的費用，直到終止或取消為止。

Console

請求 LCU 保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器名稱。
4. 在容量索引標籤上，選擇編輯 LCU 保留。
5. 選取歷史參考型估算。
6. 選取參考期間以檢視建議的預留 LCU 層級。
7. 如果您沒有歷史參考工作負載，您可以選擇手動估算，然後輸入要保留LCUs 數量。
8. 選擇儲存。

AWS CLI

請求 LCU 保留

使用 modify-capacity-reservation 命令。

```
aws elbv2 modify-capacity-reservation \
    --load-balancer-arn load-balancer-arn \
    --minimum-load-balancer-capacity CapacityUnits=100
```

CloudFormation

請求 LCU 保留

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
```

```
        - !Ref mySecurityGroup
      MinimumLoadBalancerCapacity:
        CapacityUnits: 100
```

更新或取消 Application Load Balancer 的負載平衡器容量單位保留

如果負載平衡器的流量模式變更，您可以更新或取消負載平衡器的 LCU 保留。LCU 保留的狀態必須佈建。

Console

更新或取消 LCU 保留

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器名稱。
4. 在容量索引標籤上，執行下列其中一項：
   a. 若要更新 LCU 保留，請選擇編輯 LCU 保留。
   b. 若要取消 LCU 保留，請選擇取消容量。

AWS CLI

取消 LCU 保留

使用 modify-capacity-reservation 命令。

```
aws elbv2 modify-capacity-reservation \
    --load-balancer-arn load-balancer-arn \
    --reset-capacity-reservation
```

監控 Application Load Balancer 的Application Load Balancer容量單位保留

保留狀態

以下是 LCU 保留的可能狀態值：

- pending - 表示正在進行佈建的保留。

- `provisioned` - 表示預留容量已就緒且可供使用。
- `failed` - 表示目前無法完成請求。
- `rebalancing` - 表示已新增或移除可用區域，且負載平衡器正在重新平衡容量。

LCU 使用率

`ReservedLCUs` 指標會以每分鐘為單位報告。容量會每小時保留一次。例如，如果您的 LCU 保留為 6，000，則 的一小時總計`ReservedLCUs`為 6，000，而一分鐘總計為 100。若要判斷您的預留 LCU 使用率，請參閱 `PeakLCUs` 指標。您可以設定 CloudWatch 警示來比較每分鐘 `Sum PeakLCUs`與預留容量值，或每小時 `Sum ReservedLCUs`，以判斷您是否已預留足夠的容量來滿足您的需求。

Console

檢視 LCU 保留的狀態

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器名稱。
4. 在容量索引標籤上，您可以檢視保留狀態和預留 LCU 值。

AWS CLI

監控 LCU 保留的狀態

使用 describe-capacity-reservation 命令。

```
aws elbv2 describe-capacity-reservation \
    --load-balancer-arn load-balancer-arn
```

Application Load Balancer 的整合

您可以將 Application Load Balancer 架構與數個 AWS 其他服務整合來最佳化，以增強應用程式的效能、安全性和可用性。

負載平衡器整合

- Amazon Application Recovery Controller (ARC)

- Amazon CloudFront + AWS WAF
- AWS Global Accelerator
- AWS Config
- AWS WAF

Amazon Application Recovery Controller (ARC)

Amazon Application Recovery Controller (ARC) 可協助您將負載平衡器的流量從受損的可用區域轉移到相同區域中運作狀態良好的可用區域。使用區域轉移可減少應用程式上可用區域中停電、硬體問題或軟體問題的持續時間和嚴重性。

如需詳細資訊，請參閱Application Load Balancer 的區域轉移。

Amazon CloudFront + AWS WAF

Amazon CloudFront 是一種 Web 服務，可協助改善使用的應用程式的效能、可用性和安全性 AWS。CloudFront 可做為使用 Application Load Balancer 之 Web 應用程式的分散式單一進入點。它擴展了 Application Load Balancer 的全球觸角，讓它能夠從附近的節點有效率地為使用者提供服務，最佳化內容交付並減少全球使用者的延遲。這些節點的自動內容快取可大幅降低 Application Load Balancer 的負載，進而改善其效能和可擴展性。

Elastic Load Balancing 主控台中可用的一鍵式整合會建立具有建議 AWS WAF 安全性保護的 CloudFront 分佈，並將其與您的 Application Load Balancer 建立關聯。在到達負載平衡器之前，會針對常見的 Web 漏洞 AWS WAF 進行保護。您可以從主控台的負載平衡器整合索引標籤存取 CloudFront 分佈及其對應的安全儀表板。如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的在 CloudFront 安全儀表板中管理 AWS WAF 安全保護，以及《https：// 中的 CloudFront 安全儀表板、統一 CDN 和安全體驗簡介。 aws.amazon.com/blogs

作為安全最佳實務，請將面向網際網路的 Application Load Balancer 安全群組設定為僅允許來自 CloudFront AWS受管字首清單的傳入流量，並移除任何其他傳入規則。如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》 > 中的使用 CloudFront 受管字首清單、設定 CloudFront 將自訂 HTTP 標頭新增至請求，以及設定 Application Load Balancer 僅轉送包含特定標頭的請求。Amazon CloudFront

> **Note**
>
> CloudFront 僅支援美國東部（維吉尼亞北部）us-east-1 區域的 ACM 憑證。如果您的 Application Load Balancer 在 us-east-1 以外的區域中設定了 HTTPS 接聽程式，您需要將

CloudFront 原始伺服器連線從 HTTPS 變更為 HTTP，或在美國東部（維吉尼亞北部）區域佈建 ACM 憑證，並將其連接到您的 CloudFront 分佈。

AWS Global Accelerator

若要最佳化應用程式的可用性、效能和安全性，請為您的負載平衡器建立加速器。加速器會將透過 AWS 全球網路的流量導向靜態 IP 地址，做為最接近用戶端之區域中的固定端點。 AWS Global Accelerator 受到 Shield Standard 的保護，可將 DDoS 攻擊的應用程式停機時間和延遲降至最低。

如需詳細資訊，請參閱《開發人員指南》中的在建立負載平衡器時新增加速器。 AWS Global Accelerator

AWS Config

若要最佳化負載平衡器的監控和合規，請設定 AWS Config。 AWS Config 提供 AWS 帳戶中 AWS 資源組態的詳細檢視。這包括資源彼此的關係，以及它們在過去的設定方式，以便您可以查看組態和關係如何隨著時間而變化。 AWS Config 簡化了稽核、合規和故障診斷。

如需詳細資訊，請參閱《AWS Config 開發人員指南》https://docs.aws.amazon.com/config/latest/developerguide/。

AWS WAF

您可以使用 AWS WAF 搭配 Application Load Balancer，根據 Web 存取控制清單 (Web ACL) 中的規則來允許或封鎖請求。

根據預設，如果負載平衡器無法從中取得回應 AWS WAF，它會傳回 HTTP 500 錯誤，而且不會轉送請求。如果您需要負載平衡器將請求轉送到目標 AWS WAF，即使無法聯絡，也可以啟用 AWS WAF 失敗開啟。

預先定義的 Web ACLs

啟用 AWS WAF 整合時，您可以選擇使用預先定義的規則自動建立新的 Web ACL。預先定義的 Web ACL 包含三個 AWS 受管規則，可針對最常見的安全威脅提供保護。

- `AWSManagedRulesAmazonIpReputationList` - Amazon IP 評價清單規則群組會封鎖通常與機器人或其他威脅相關聯的 IP 地址。如需詳細資訊，請參閱《 AWS WAF 開發人員指南》中的 Amazon IP 評價清單受管規則群組。

- `AWSManagedRulesCommonRuleSet` - 核心規則集 (CRS) 規則群組提供保護，防止利用各種漏洞，包括 OWASP 出版品中所述的一些高風險和常見漏洞，例如 OWASP 前 10 名。如需詳細資訊，請參閱《 AWS WAF 開發人員指南》中的核心規則集 (CRS) 受管規則群組。
- `AWSManagedRulesKnownBadInputsRuleSet` - 已知錯誤輸入規則群組會封鎖已知無效的請求模式，並與漏洞的利用或探索相關聯。如需詳細資訊，請參閱《 AWS WAF 開發人員指南》中的已知錯誤輸入受管規則群組。

如需詳細資訊，請參閱《 AWS WAF 開發人員指南》中的在 中使用 Web ACLs AWS WAF。

Application Load Balancer 的接聽程式

接聽程式是檢查連線請求的程序，必須使用您已設定的通訊協定與連接埠。開始使用 Application Load Balancer 之前，必須新增至少一個接聽程式。如果負載平衡器沒有接聽程式，就無法接收來自用戶端的流量。您為接聽程式定義的規則，將決定負載平衡器將請求路由到已註冊目標 (例如 EC2 執行個體) 的方法。

目錄

- 接聽程式組態
- 接聽程式屬性
- 預設動作
- 為 Application Load Balancer 建立 HTTP 接聽程式
- Application Load Balancer 的 SSL 憑證
- Application Load Balancer 的安全政策
- 為 Application Load Balancer 建立 HTTPS 接聽程式
- 為 Application Load Balancer 更新 HTTPS 接聽程式
- Application Load Balancer 的接聽程式規則
- 在 Application Load Balancer 中使用 TLS 進行相互身分驗證
- 使用 Application Load Balancer 來驗證使用者身分
- 使用 Application Load Balancer 驗證 JWTs
- HTTP 標頭和 Application Load Balancer
- Application Load Balancer 的 HTTP 標頭修改
- 刪除 Application Load Balancer 的接聽程式

接聽程式組態

接聽程式支援下列通訊協定與連接埠：

- Protocols (通訊協定)：HTTP、HTTPS
- Ports (連接埠)：1-65535

您可以使用 HTTPS 接聽程式來將加密和解密的工作卸載到您的負載平衡器，使得您的應用程式可以專注在商業邏輯上。如果接聽程式通訊協定是 HTTPS，則必須在接聽程式上至少部署一個 SSL 伺服器憑證。如需詳細資訊，請參閱為 Application Load Balancer 建立 HTTPS 接聽程式。

如果您必須確保目標解密的是 HTTPS 流量 (而不是負載平衡器)，則可以建立在連接埠 443 上具有 TCP 接聽程式的 Network Load Balancer。使用 TCP 接聽程式時，負載平衡器會將加密的流量傳遞給目標，而不需要對流量進行解密。如需詳細資訊，請參閱《Network Load Balancer 使用者指南》。

WebSockets

Application Load Balancer 提供對 WebSocket 的原生支援。您可以使用 HTTP 連線，升級現有的與 WebSocket (`ws` 或 `wss`) 的 HTTP/1.1 連線。升級時，用於請求 (對象是負載平衡器以及目標) 的 TCP 連線會透過負載平衡器變成用戶端與目標之間的持續性 WebSocket 連線。您可以透過 HTTP 和 HTTPS 接聽程式來使用 WebSocket。您為接聽程式選擇的選項會套用到 WebSocket 連線和 HTTP 流量。路由至已啟用目標最佳化工具之目標群組的請求不支援 Websocket。如需詳細資訊，請參閱《Amazon CloudFront 開發人員指南》中的 WebSocket 通訊協定的運作方式。

HTTP/2

Application Load Balancer 對使用 HTTPS 接聽程式的 HTTP/2 提供原生支援。您可以使用 HTTP/2 連線平行傳送高達 128 個請求。您可以使用通訊協定版本，使用 HTTP/2 將請求傳送至目標。如需詳細資訊，請參閱通訊協定版本。由於 HTTP/2 可更有效率地使用前端連線，您可能會注意到用戶端和負載平衡器之間的連線數較少。您無法使用 HTTP/2 的伺服器推送功能。

Application Load Balancer 的相互 TLS 身分驗證在傳遞和驗證模式下都支援 HTTP/2。如需詳細資訊，請參閱在 Application Load Balancer 中使用 TLS 進行相互身分驗證。

如需詳細資訊，請參閱 Elastic Load Balancing User Guide 中的 Request routing。

接聽程式屬性

以下是 Application Load Balancer 的接聽程式屬性：

`routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name`

可讓您修改 X-Amzn-Mtls-Clientcert-Serial-Number HTTP 請求標頭的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_issuer.header_name`

可讓您修改 X-Amzn-Mtls-Clientcert-Issuer HTTP 請求標頭的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_subject.header_name`

可讓您修改 X-Amzn-Mtls-Clientcert-Subject HTTP 請求標頭的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_validity.header_name`

可讓您修改 X-Amzn-Mtls-Clientcert-Validity HTTP 請求標頭的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_leaf.header_name`

可讓您修改 X-Amzn-Mtls-Clientcert-Leaf HTTP 請求標頭的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert.header_name`

可讓您修改 X-Amzn-Mtls-Clientcert HTTP 請求標頭的標頭名稱。

`routing.http.request.x_amzn_tls_version.header_name`

可讓您修改 X-Amzn-Tls-Version HTTP 請求標頭的標頭名稱。

`routing.http.request.x_amzn_tls_cipher_suite.header_name`

可讓您修改 X-Amzn-Tls-Cipher-Suite HTTP 請求標頭的標頭名稱。

`routing.http.response.server.enabled`

可讓您允許或移除 HTTP 回應伺服器標頭。

`routing.http.response.strict_transport_security.header_value`

通知瀏覽器應該只使用 HTTPS 存取網站，而且未來任何使用 HTTP 存取網站的嘗試都應該自動轉換為 HTTPS。

`routing.http.response.access_control_allow_origin.header_value`

指定允許存取伺服器的原始伺服器。

`routing.http.response.access_control_allow_methods.header_value`

傳回從不同原始伺服器存取伺服器時，允許哪些 HTTP 方法。

`routing.http.response.access_control_allow_headers.header_value`

指定在請求期間可以使用哪些標頭。

`routing.http.response.access_control_allow_credentials.header_value`

指出提出請求時，瀏覽器是否應包含登入資料，例如 Cookie 或身分驗證。

`routing.http.response.access_control_expose_headers.header_value`

傳回瀏覽器可以向請求用戶端公開的標頭。

`routing.http.response.access_control_max_age.header_value`

指定預檢請求結果的快取時間，以秒為單位。

`routing.http.response.content_security_policy.header_value`

指定瀏覽器強制執行的限制，以協助將特定類型安全威脅的風險降至最低。

`routing.http.response.x_content_type_options.header_value`

指出是否應遵循且不變更 Content-Type 標頭中公告的 MIME 類型。

`routing.http.response.x_frame_options.header_value`

指出是否允許瀏覽器轉譯影格、iframe、內嵌或物件中的頁面。

預設動作

每個接聽程式都有預設動作，也稱為預設規則。無法刪除預設規則，且一律最後執行。您可以建立其他規則。這些規則包含優先順序、一或多個動作，以及一或多個條件。您可以隨時新增或編輯規則。如需詳細資訊，請參閱接聽程式規則。

為 Application Load Balancer 建立 HTTP 接聽程式

接聽程式會檢查連線請求。當您在立負載平衡器時便定義接聽程式，然後可隨時新增接聽程式到您的負載平衡器。

此頁面的資訊協助您為負載平衡器建立 HTTP 接聽程式。若要將 HTTPS 接聽程式新增至您的負載平衡器，請參閱為 Application Load Balancer 建立 HTTPS 接聽程式。

先決條件

- 若要新增轉送動作到預設的接聽程式規則，您必須指定可用的目標群組。如需詳細資訊，請參閱為您的 Application Load Balancer 建立目標群組。
- 您可以在多個接聽程式中指定相同的目標群組，但這些接聽程式必須屬於相同的負載平衡器。若要將目標群組與負載平衡器搭配使用，您必須確認沒有其他負載平衡器的接聽程式使用該目標群組。

新增 HTTP 接聽程式

您使用用戶端與負載平衡器間連線的通訊協定與連接埠來設定接聽程式，並為預設接聽程式規則設定目標群組。如需詳細資訊，請參閱接聽程式組態。

若要新增另一個接聽程式規則，請參閱 接聽程式規則。

Console

新增 HTTP 接聽程式

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選擇新增接聽程式。
5. 針對通訊協定，選擇 HTTP。保留預設連接埠或輸入不同的連接埠。
6. 針對預設動作，選取下列其中一個路由動作，並提供必要資訊：
   - 轉送至目標群組 – 選擇目標群組。若要新增另一個目標群組，請選擇新增目標群組、選擇目標群組、檢閱相對權重，並視需要更新權重。如果您在任何目標群組上啟用粘性，則必須啟用群組層級粘性。

     如果您沒有符合您需求的目標群組，請選擇建立目標群組以立即建立目標群組。如需詳細資訊，請參閱建立目標群組。
   - 重新導向至 URL – 在 URI 部分索引標籤上分別輸入每個部分，或在完整 URL 索引標籤上輸入完整地址，以輸入 URL。針對狀態碼，根據您的需求選取暫時 (HTTP 302) 或永久 (HTTP 301)。
   - 傳回固定回應 – 輸入回應碼以針對捨棄的用戶端請求傳回。或者，您可以指定內容類型和回應內文。
7. （選用）若要新增標籤，請展開接聽程式標籤。選擇新增標籤，然後輸入標籤索引鍵和標籤值。
8. 選擇 Add listener (新增接聽程式)。

AWS CLI

若要建立目標群組

如果您沒有可用於預設動作的目標群組，請使用 create-target-group 命令立即建立一個。如需範例，請參閱 建立目標群組。

建立 HTTP 接聽程式

使用 create-listener 命令。下列範例會建立 HTTP 接聽程式，其中包含將流量轉送至指定目標群組的預設規則。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
    --protocol HTTP \
    --port 80 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn
```

若要建立在兩個目標群組之間分配流量的轉送動作，請改用下列--default-actions選項。指定多個目標群組時，您必須為每個目標群組提供權重。

```
    --default-actions '[{
          "Type":"forward",
          "ForwardConfig":{
            "TargetGroups":[
              {"TargetGroupArn":"target-group-1-arn","Weight":50},
              {"TargetGroupArn":"target-group-2-arn","Weight":50}
            ]
          }
        }]'
```

CloudFormation

建立 HTTP 接聽程式

定義 AWS::ElasticLoadBalancingV2::Listener 類型的資源。下列範例會建立 HTTP 接聽程式，其中包含將流量轉送至指定目標群組的預設規則。

```
Resources:
  myHTTPlistener:
  Type: 'AWS::ElasticLoadBalancingV2::Listener'
  Properties:
    LoadBalancerArn: !Ref myLoadBalancer
    Protocol: HTTP
    Port: 80
```

```
    DefaultActions:
      - Type: "forward"
        TargetGroupArn: !Ref myTargetGroup
```

若要建立在多個目標群組之間分配流量的轉送動作，請使用 ForwardConfig 屬性。指定多個目標群組時，您必須為每個目標群組提供權重。

```
Resources:
  myHTTPlistener:
  Type: 'AWS::ElasticLoadBalancingV2::Listener'
  Properties:
    LoadBalancerArn: !Ref myLoadBalancer
    Protocol: HTTP
    Port: 80
    DefaultActions:
      - Type: "forward"
        ForwardConfig:
          TargetGroups:
            - TargetGroupArn: !Ref TargetGroup1
              Weight: 50
            - TargetGroupArn: !Ref TargetGroup2
              Weight: 50
```

Application Load Balancer 的 SSL 憑證

當您為 Application Load Balancer 建立安全接聽程式時，您必須在負載平衡器上部署至少一個憑證。負載平衡器需要 X.509 憑證 (SSL/TLS 伺服器憑證)。憑證為憑證授權機構 (CA) 發出的數位形式身分證明。憑證包含識別資訊、有效期間、公有金鑰、序號和發行者的數位簽章。

建立憑證以搭配您的負載平衡器使用時，您必須指定網域名稱。憑證上的網域名稱必須與自訂網域名稱記錄相符，如此我們就可以確認 TLS 連線。如果其不相符，就不會加密流量。

您必須為憑證指定完整網域名稱 (FQDN)，例如 www.example.com；或者指定 apex 網域名稱 (FQDN)，例如 example.com。您也可以使用星號 (*) 做為萬用字元，以保護相同網域中的多個網站名稱。請求萬用字元憑證時，星號 (*) 必須在網域名稱的最左方，而且僅能保護一個子網域層級。例如，*.example.com 保護 corp.example.com 和 images.example.com，但它無法保護 test.login.example.com。另請注意，*.example.com 只可以保護 example.com 的子網域，無法保護 bare 或 apex 網域 (example.com)。萬用字元名稱會顯示於憑證的主體欄位和主體別名延伸。如需公有憑證的詳細資訊，請參閱AWS Certificate Manager 《 使用者指南》中的請求公有憑證。

建議您使用 AWS Certificate Manager (ACM) 為負載平衡器建立憑證。ACM 支援具有 2048、3072 和 4096 位元金鑰長度的 RSA 憑證，以及所有 ECDSA 憑證。ACM 會與 Elastic Load Balancing 整合，以便您在負載平衡器上部署憑證。如需詳細資訊，請參閱「AWS Certificate Manager 使用者指南」。

或者，您可以使用 SSL/TLS 工具來建立憑證簽署請求 (CSR)，然後取得 CA 簽署的 CSR 來產生憑證，然後將憑證匯入 ACM 或上傳憑證至 AWS Identity and Access Management (IAM)。如需有關將憑證匯入 ACM 的詳細資訊，請參閱《AWS Certificate Manager 使用者指南》中的匯入憑證。如需上傳憑證至 IAM 的詳細資訊，請參閱 IAM 使用者指南中的使用伺服器憑證。

預設憑證

建立 HTTPS 接聽程式時，您必須指定剛好一個憑證。此憑證稱為預設憑證。您可以在建立 HTTPS 接聽程式之後取代預設憑證。如需詳細資訊，請參閱更換預設憑證。

如果您在憑證清單中指定額外憑證，只有當用戶端連接時未使用伺服器名稱指示 (SNI) 通訊協定來指定主機名稱，或憑證清單中沒有相符的憑證時，才會使用預設憑證。

如果您不指定額外憑證，但需要透過單一負載平衡器來託管多個安全應用程式，您可以使用萬用字元憑證，或將每個額外網域的主體別名 (SAN) 新增至憑證。

憑證清單

建立 HTTPS 接聽程式之後，您可以將憑證新增至憑證清單。如果您使用 建立接聽程式 AWS 管理主控台，我們會將預設憑證新增至憑證清單。否則，憑證清單為空白。使用憑證清單可讓負載平衡器在相同連接埠上支援多個網域，並為每個網域提供不同的憑證。如需詳細資訊，請參閱將憑證新增至憑證清單。

負載平衡器使用支援 SNI 的智慧憑證選擇演算法。如果用戶端提供的主機名稱符合憑證清單中的單一憑證，負載平衡器會選取此憑證。如果用戶端提供的主機名稱符合憑證清單中的多個憑證，負載平衡器會選取用戶端可支援的最佳憑證。憑證選擇是根據採用下列順序的以下條件：

- 公有金鑰演算法 (ECDSA 優於 RSA)
- 過期 （偏好未過期）
- 雜湊演算法 （偏好 SHA 而非 MD5)。如果有多個 SHA 憑證，則偏好最高的 SHA 號碼。
- 金鑰長度 (最好是最大)
- 有效期間

負載平衡器存取日誌項目會指出用戶端指定的主機名稱和向用戶端出示的憑證。如需詳細資訊，請參閱存取日誌項目。

憑證續約

每個憑證均附帶有效期間。您必須確保在有效期間結束之前，續約或更換負載平衡器的每個憑證。這包括預設憑證和憑證清單中的憑證。續約或更換憑證不會影響負載平衡器節點收到並且等待路由到運作狀態良好目標的傳輸中請求。續約憑證之後，新請求會使用續約的憑證。更換憑證之後，新請求會使用新的憑證。

您可以如下所示管理憑證續約和更換：

- 負載平衡器上提供 AWS Certificate Manager 和部署的憑證可以自動續約。ACM 會在憑證過期之前嘗試續約。如需詳細資訊，請參閱 AWS Certificate Manager 使用者指南中的受管續約。
- 如果您將憑證匯入至 ACM，則必須監控憑證的過期日期，並在憑證過期之前續約。如需詳細資訊，請參閱 AWS Certificate Manager 使用者指南中的匯入憑證。
- 如果您將憑證匯入至 IAM，則必須建立新的憑證、將新的憑證匯入至 ACM 或 IAM、將新憑證新增至負載平衡器，並從負載平衡器移除過期的憑證。

Application Load Balancer 的安全政策

Elastic Load Balancing 使用 Secure Sockets Layer (SSL) 交涉組態 (稱為安全政策)，在用戶端與負載平衡器之間交涉 SSL 連線。安全政策為通訊協定與加密的組合。通訊協定會在用戶端和伺服器之間建立安全連線，並確保用戶端和負載平衡器之間傳遞的所有資料都是私有的。密碼是一種加密演算法，使用加密金鑰來建立編碼的訊息。通訊協定使用多個密碼來加密網際網路上的資料。在連線交涉程序期間，用戶端與負載平衡器會出示它們分別支援的加密和通訊協定的清單 (以偏好的順序)。在預設情況下，將針對安全連線選取伺服器清單上符合任何用戶端加密的第一個加密。

考量事項

- HTTPS 接聽程式需要安全政策。如果您在建立接聽程式時未指定安全政策，我們會使用預設安全政策。預設安全政策取決於您建立 HTTPS 接聽程式的方式：
  - 主控台 – 預設安全政策為 `ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09`。
  - 其他方法 （例如 AWS CLI AWS CloudFormation和 AWS CDK) – 預設安全政策為 `ELBSecurityPolicy-2016-08`。
  - 若要檢視連線請求至負載平衡器的 TLS 通訊協定版本 （日誌欄位位置 5) 和金鑰交換 （日誌欄位位置 13)，請啟用連線記錄並檢查對應的日誌項目。如需詳細資訊，請參閱連線日誌。
  - 名稱中具有 PQ 的安全政策提供混合式後量子金鑰交換。為了相容性，它們支援傳統和後量子 ML-KEM 金鑰交換演算法。用戶端必須支援 ML-KEM 金鑰交換，才能使用混合式後量子 TLS

進行金鑰交換。混合式後量子政策支援 SecP256r1MLKEM768, SecP384r1MLKEM1024和 X25519MLKEM768 演算法。如需詳細資訊，請參閱後量子密碼編譯。

  - AWS 建議實作新的後量子 TLS (PQ-TLS) 型安全政策ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09或 ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09。此政策透過支援僅能夠交涉混合 PQ-TLS、TLS 1.3 或 TLS 1.2 的用戶端來確保回溯相容性，從而最大限度地減少轉換為量子後密碼編譯期間的服務中斷。隨著用戶端應用程式開發交涉 PQ-TLS 以進行金鑰交換操作的能力，您可以逐步遷移到更嚴格的安全政策。

- 若要符合需要停用特定 TLS 通訊協定版本的合規和安全標準，或支援需要已棄用加密的舊版用戶端，您可以使用其中一個ELBSecurityPolicy-TLS-安全政策。若要檢視 Application Load Balancer 請求的 TLS 通訊協定版本，請啟用負載平衡器的存取記錄，並檢查對應的存取日誌項目。如需詳細資訊，請參閱存取日誌。
- 您可以分別在 IAM 和服務控制政策 (SCPs) 中使用 Elastic Load Balancing 條件金鑰 AWS 帳戶 ，來限制哪些安全政策可供 AWS Organizations 和 的使用者使用。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策 (SCP)。
- 僅支援 TLS 1.3 的政策支援轉送秘密 (FS)。支援僅具有 TLS_* 和 ECDHE_* 格式密碼的 TLS 1.3 和 TLS 1.2 的政策也提供 FS。
- Application Load Balancer 支援使用 PSK (TLS 1.3) 和工作階段 IDs/工作階段票證 (TLS 1.2 及更舊版本） 的 TLS 恢復。只有在連線到相同的 Application Load Balancer IP 地址時，才支援恢復。0-RTT 資料功能和 early_data 延伸未實作。
- Application Load Balancers 不支援自訂安全政策。
- Application Load Balancers 僅支援目標連線的 SSL 重新交涉。

後端連線

- 您可以選擇用於前端連線的安全政策，但不能選擇後端連線。後端連線的安全政策取決於接聽程式安全政策。如果有任何接聽程式正在使用：
  - FIPS 後量子 TLS 政策 - 後端連線使用 ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09
  - FIPS 政策 - 後端連線使用 ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04
  - 後量子 TLS 政策 - 後端連線使用 ELBSecurityPolicy-TLS13-1-0-PQ-2025-09
  - TLS 1.3 政策 - 後端連線使用 ELBSecurityPolicy-TLS13-1-0-2021-06
  - 其他 TLS 政策 - 後端連線使用 ELBSecurityPolicy-2016-08

安全政策

- describe-ssl-policies 命令範例
- TLS 安全政策
  - 依政策的通訊協定
  - 依政策的 Ciphers
  - 依密碼排列的政策
- FIPS 安全政策
  - 依政策的通訊協定
  - 依政策的 Ciphers
  - 依密碼排列的政策
- FS 支援的政策
  - 依政策的通訊協定
  - 依政策的 Ciphers
  - 依密碼排列的政策

describe-ssl-policies 命令範例

您可以使用 describe-ssl-policies AWS CLI 命令，描述安全政策的通訊協定和密碼，或尋找符合您需求的政策。

下列範例說明指定的政策。

```
aws elbv2 describe-ssl-policies \
    --names "ELBSecurityPolicy-TLS13-1-2-Res-2021-06"
```

下列範例列出政策名稱中具有指定字串的政策。

```
aws elbv2 describe-ssl-policies \
    --query "SslPolicies[?contains(Name,'FIPS')].Name"
```

下列範例列出支援指定通訊協定的政策。

```
aws elbv2 describe-ssl-policies \
    --query "SslPolicies[?contains(SslProtocols,'TLSv1.3')].Name"
```

下列範例列出支援指定密碼的政策。

```
aws elbv2 describe-ssl-policies \
    --query "SslPolicies[?Ciphers[?contains(Name,'TLS_AES_128_GCM_SHA256')]].Name"
```

下列範例列出不支援指定密碼的政策。

```
aws elbv2 describe-ssl-policies \
    --query 'SslPolicies[?length(Ciphers[?starts_with(Name,`AES128-GCM-SHA256`)]) ==
 `0`].Name'
```

TLS 安全政策

您可以使用 TLS 安全政策來符合需要停用特定 TLS 通訊協定版本的合規和安全標準，或支援需要已棄用密碼的舊版用戶端。

僅支援 TLS 1.3 的政策支援轉送秘密 (FS)。支援僅具有 TLS_* 和 ECDHE_* 格式密碼的 TLS 1.3 和 TLS 1.2 的政策也提供 FS。

目錄

- 依政策的通訊協定
- 依政策的 Ciphers
- 依密碼排列的政策

依政策的通訊協定

下表說明每個 TLS 安全政策支援的通訊協定。

安全政策	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-3-2021-06	是	否	否	否
ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	是	否	否	否
ELBSecurityPolicy-TLS13-1-2-2021-06	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Res-2021-06	是	是	否	否

安全政策	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-1-2021-06	是	是	是	否
ELBSecurityPolicy-TLS13-1-0-2021-06	是	是	是	是
ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	是	是	是	是
ELBSecurityPolicy-TLS-1-2-Ext-2018-06	否	是	否	否
ELBSecurityPolicy-TLS-1-2-2017-01	否	是	否	否
ELBSecurityPolicy-TLS-1-1-2017-01	否	是	是	否
ELBSecurityPolicy-2016-08	否	是	是	是

依政策的 Ciphers

下表說明每個 TLS 安全政策支援的加密。

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-3-2021-06 ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256
ELBSecurityPolicy-TLS13-1-2-2021-06 ELBSecurityPolicy-TLS13-1-2-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384

安全政策	加密方式
	• TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-TLS13-1-2-Res-2021-06 ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-1-2021-06	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-0-2021-06 ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-TLS-1-2-Ext-2018-06	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-TLS-1-2-2017-01	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256

安全政策	加密方式
ELBSecurityPolicy-TLS-1-1-2017-01	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-2016-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

依密碼排列的政策

下表說明支援每個密碼的 TLS 安全政策。

密碼名稱	安全政策	密碼套件
OpenSSL – TLS_AES_128_GCM_SHA256 IANA – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	1301

密碼名稱	安全政策	密碼套件
	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	

密碼名稱	安全政策	密碼套件
OpenSSL – TLS_AES_256_GCM_SHA384 IANA – TLS_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	1302

密碼名稱	安全政策	密碼套件
OpenSSL – TLS_CHACHA20_POLY1305_SHA256 IANA – TLS_CHA20_POLY1305_SHA256	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	1303

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02b

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02f

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES128-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c023

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES128-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c027

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES128-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c009
OpenSSL – ECDHE-RSA-AES128-SHA IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c013

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02c

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c030

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES256-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c024

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES256-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c028

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES256-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c00a
OpenSSL – ECDHE-RSA-AES256-SHA IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c014

密碼名稱	安全政策	密碼套件
OpenSSL – AES128-GCM-SHA256 IANA – TLS_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	9c

密碼名稱	安全政策	密碼套件
OpenSSL – AES128-SHA256 IANA – TLS_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	3c

密碼名稱	安全政策	密碼套件
OpenSSL – AES128-SHA IANA – TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	2f

密碼名稱	安全政策	密碼套件
OpenSSL – AES256-GCM-SHA384 IANA – TLS_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	9d

密碼名稱	安全政策	密碼套件
OpenSSL – AES256-SHA256 IANA – TLS_RSA_WITH_AES_256_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	3d

密碼名稱	安全政策	密碼套件
OpenSSL – AES256-SHA IANA – TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	35

FIPS 安全政策

聯邦資訊處理標準 (FIPS) 是美國和加拿大政府標準，指定保護敏感資訊之密碼編譯模組的安全要求。若要進一步了解，請參閱AWS 雲端安全合規頁面上的聯邦資訊處理標準 (FIPS) 140。

所有 FIPS 政策都會利用 AWS-LC FIPS 驗證的密碼編譯模組。若要進一步了解，請參閱 NIST 密碼編譯模組驗證計劃網站上的 AWS-LC 密碼編譯模組頁面。

> **Important**
>
> 政策和 `ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04` `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04` 僅供舊版相容性使用。雖然他們使用 FIPS140 模組來使用 FIPS 密碼編譯，但可能不符合 TLS 組態的最新 NIST 指引。

目錄

- 依政策的通訊協定
- 依政策的 Ciphers

- 依密碼排列的政策

依政策的通訊協定

下表說明每個 FIPS 安全政策支援的通訊協定。

安全政策	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04	是	否	否	否
ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09	是	否	否	否
ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04	是	是	否	否
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09	是	是	否	否
ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04	是	是	是	否
ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04	是	是	是	是
ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	是	是	是	是

依政策的 Ciphers

下表說明每個 FIPS 安全政策支援的加密。

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384
ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256

安全政策	加密方式
	• ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

依密碼排列的政策

下表說明支援每個密碼的 FIPS 安全政策。

密碼名稱	安全政策	密碼套件
OpenSSL – TLS_AES_128_GCM_SHA256 IANA – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04	1301

密碼名稱	安全政策	密碼套件
	• ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	

密碼名稱	安全政策	密碼套件
OpenSSL – TLS_AES_256_GCM_SHA384 IANA – TLS_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	1302

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02b

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02f

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES128-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c023

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES128-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c027

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES128-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c009
OpenSSL – ECDHE-RSA-AES128-SHA IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c013

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02c

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c030

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES256-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c024

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES256-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c028

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES256-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c00a
OpenSSL – ECDHE-RSA-AES256-SHA IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c014

密碼名稱	安全政策	密碼套件
OpenSSL – AES128-GCM-SHA256 IANA – TLS_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	9c
OpenSSL – AES128-SHA256 IANA – TLS_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	3c

密碼名稱	安全政策	密碼套件
OpenSSL – AES128-SHA IANA – TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	2f
OpenSSL – AES256-GCM-SHA384 IANA – TLS_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	9d

密碼名稱	安全政策	密碼套件
OpenSSL – AES256-SHA256 IANA – TLS_RSA_WITH_AES_256_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	3d
OpenSSL – AES256-SHA IANA – TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	35

FS 支援的政策

FS (Forward Secrecy) 支援的安全政策透過使用唯一的隨機工作階段金鑰，提供額外的保護，防止加密資料的竊聽。這可防止對擷取的資料進行解碼，即使秘密長期金鑰遭到入侵也一樣。

本節中的政策支援 FS，且「FS」包含在其名稱中。不過，這些並非支援 FS 的唯一政策。僅支援 TLS 1.3 的政策支援 FS。支援僅具有 TLS_* 和 ECDHE_* 格式密碼的 TLS 1.3 和 TLS 1.2 的政策也提供 FS。

目錄

- 依政策的通訊協定
- 依政策的 Ciphers
- 依密碼排列的政策

依政策的通訊協定

下表說明每個 FS 支援的安全政策支援的通訊協定。

安全政策	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-FS-1-2-Res-2020-10	否	是	否	否
ELBSecurityPolicy-FS-1-2-Res-2019-08	否	是	否	否
ELBSecurityPolicy-FS-1-2-2019-08	否	是	否	否
ELBSecurityPolicy-FS-1-1-2019-08	否	是	是	否
ELBSecurityPolicy-FS-2018-06	否	是	是	是

依政策的 Ciphers

下表說明每個 FS 支援的安全政策支援的密碼。

安全政策	加密方式
ELBSecurityPolicy-FS-1-2-Res-2020-10	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384

安全政策	加密方式
ELBSecurityPolicy-FS-1-2-Res-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-FS-1-2-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

安全政策	加密方式
ELBSecurityPolicy-FS-1-1-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA
ELBSecurityPolicy-FS-2018-06	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

依密碼排列的政策

下表說明支援每個密碼的 FS 支援安全政策。

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02b
OpenSSL – ECDHE-RSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02f
OpenSSL – ECDHE-ECDSA-AES128-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c023
OpenSSL – ECDHE-RSA-AES128-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c027
OpenSSL – ECDHE-ECDSA-AES128-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c009

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-RSA-AES128-SHA IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c013
OpenSSL – ECDHE-ECDSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02c
OpenSSL – ECDHE-RSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c030
OpenSSL – ECDHE-ECDSA-AES256-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c024
OpenSSL – ECDHE-RSA-AES256-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c028

密碼名稱	安全政策	密碼套件
OpenSSL – ECDHE-ECDSA-AES256-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c00a
OpenSSL – ECDHE-RSA-AES256-SHA IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c014

為 Application Load Balancer 建立 HTTPS 接聽程式

接聽程式會檢查連線請求。當您在立負載平衡器時便定義接聽程式，然後可隨時新增接聽程式到您的負載平衡器。

若要建立 HTTPS 接聽程式，您必須在負載平衡器上部署至少一個 SSL 伺服器憑證。負載平衡器使用伺服器憑證終止前端連接，然後解密用戶端的請求，再將它們傳送到目標。您還必須指定安全政策，用於交涉用戶端和負載平衡器之間的安全連線。

如果您需要將加密流量傳遞給目標，而不需要負載平衡器解密流量，就可以建立在連接埠 443 上具有 TCP 接聽程式的 Network Load Balancer 或 Classic Load Balancer。使用 TCP 接聽程式時，負載平衡器會將加密的流量傳遞給目標，而不需要對流量進行解密。

此頁面的資訊協助您為負載平衡器建立 HTTPS 接聽程式。若要將 HTTP 接聽程式新增至您的負載平衡器，請參閱為 Application Load Balancer 建立 HTTP 接聽程式。

先決條件

- 若要新增轉送動作到預設的接聽程式規則，您必須指定可用的目標群組。如需詳細資訊，請參閱為您的 Application Load Balancer 建立目標群組。
- 您可以在多個接聽程式中指定相同的目標群組，但這些接聽程式必須屬於相同的負載平衡器。若要將目標群組與負載平衡器搭配使用，您必須確認沒有其他負載平衡器的接聽程式使用該目標群組。
- Application Load Balancer 不支援 ED25519 金鑰。

新增 HTTPS 接聽程式

您可以使用通訊協定和連接埠來設定接聽程式，以便從用戶端連線至負載平衡器。如需詳細資訊，請參閱接聽程式組態。

建立安全接聽程式時，您必須指定安全政策和憑證。若要將憑證新增至憑證清單，請參閱 the section called “將憑證新增至憑證清單”。

您必須為接聽程式設定預設規則。您可以在建立接聽程式之後新增其他接聽程式規則。如需詳細資訊，請參閱接聽程式規則。

Console

新增 HTTPS 接聽程式

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選擇新增接聽程式。
5. 請在 Protocol (通訊協定) 中選擇 HTTPS。保留預設連接埠或輸入不同的連接埠。
6. （選用）針對預先路由動作，選取下列其中一個動作：
   - 驗證使用者 – 選擇身分提供者並提供必要的資訊。如需詳細資訊，請參閱使用 Application Load Balancer 來驗證使用者身分。
   - 驗證字符 – 輸入 JWKS 端點、問題和任何其他宣告。如需詳細資訊，請參閱使用 Application Load Balancer 驗證 JWTs。
7. 針對路由動作，選取下列其中一個動作：
   - 轉送至目標群組 – 選擇目標群組。若要新增另一個目標群組，請選擇新增目標群組、選擇目標群組、檢閱相對權重，並視需要更新權重。如果您在任何目標群組上啟用粘性，則必須啟用群組層級粘性。

     如果您沒有符合您需求的目標群組，請選擇建立目標群組以立即建立目標群組。如需詳細資訊，請參閱建立目標群組。
   - 重新導向至 URL – 在 URI 部分索引標籤上分別輸入每個部分，或在完整 URL 索引標籤上輸入完整地址，以輸入 URL。針對狀態碼，根據您的需求選取暫時 (HTTP 302) 或永久 (HTTP 301)。

- 傳回固定回應 – 輸入回應碼以針對捨棄的用戶端請求傳回。或者，您可以指定內容類型和回應內文。

8. 針對安全政策，我們會選取建議的安全政策。您可以視需要選取不同的安全政策。
9. 針對預設 SSL/TLS 憑證，選擇預設憑證。我們也會將預設憑證新增至 SNI 清單。您可以使用下列其中一個選項來選取憑證：

   - 從 ACM – 從憑證（從 ACM) 選擇憑證，以顯示可用的憑證 AWS Certificate Manager。
   - 從 IAM – 從憑證（從 IAM) 選擇憑證，以顯示您匯入的憑證 AWS Identity and Access Management。
   - 匯入憑證 – 選擇憑證的目的地；匯入至 ACM 或匯入至 IAM。對於憑證私有金鑰，請複製並貼上私有金鑰檔案 (PEM 編碼）的內容。對於憑證內文，複製並貼上公有金鑰憑證檔案 (PEM 編碼）的內容。對於憑證鏈，請複製並貼上憑證鏈檔案 (PEM 編碼）的內容，除非您使用的是自我簽署憑證，而且瀏覽器不一定要隱含地接受憑證。

10. （選用）若要啟用交互身分驗證，請在用戶端憑證處理下，啟用相互身分驗證 (mTLS)。

    預設模式為傳遞。如果您選取使用信任存放區驗證：

    - 根據預設，具有過期用戶端憑證的連線會遭到拒絕。若要變更此行為，請展開進階 mTLS 設定，然後在用戶端憑證過期下，選取允許過期的用戶端憑證。
    - 針對信任存放區，選擇現有的信任存放區，或選擇新增信任存放區並提供必要的資訊。

11. （選用）若要新增標籤，請展開接聽程式標籤。選擇新增標籤，然後輸入標籤索引鍵和標籤值。
12. 選擇 Add listener (新增接聽程式)。

AWS CLI

建立 HTTPS 接聽程式

使用 create-listener 命令。下列範例會建立具有預設規則的 HTTPS 接聽程式，將流量轉送至指定的目標群組。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
    --protocol HTTPS \
    --port 443 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn \
    --ssl-policy ELBSecurityPolicy-TLS13-1-2-2021-06 \
```

```
    --certificates certificate-arn
```

CloudFormation

建立 HTTPS 接聽程式

定義 AWS::ElasticLoadBalancingV2::Listener 類型的資源。下列範例會建立具有預設規則的 HTTPS 接聽程式，將流量轉送至指定的目標群組。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
      Certificates:
        - CertificateArn: certificate-arn
```

為 Application Load Balancer 更新 HTTPS 接聽程式

建立 HTTPS 接聽程式之後，您可以更換預設憑證、更新憑證清單或更換安全政策。

任務

- 更換預設憑證
- 將憑證新增至憑證清單
- 從憑證清單中移除憑證
- 更新安全政策
- HTTP 標頭修改

更換預設憑證

您可以使用以下程序，更換接聽程式的預設憑證。如需詳細資訊，請參閱預設憑證。

Console

若要取代預設憑證

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選擇通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 在憑證索引標籤上，選擇變更預設值。
6. 在 ACM 和 IAM 憑證資料表中，選取新的預設憑證。
7. （選用）根據預設，我們會選取將先前的預設憑證新增至接聽程式憑證清單。我們建議您保持選取此選項，除非您目前沒有 SNI 的接聽程式憑證，並依賴 TLS 工作階段恢復。
8. 選擇儲存為預設。

AWS CLI

若要取代預設憑證

使用 modify-listener 命令。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --certificates CertificateArn=new-default-certificate-arn
```

CloudFormation

若要取代預設憑證

更新 AWS::ElasticLoadBalancingV2::Listener。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
```

```
      - Type: "forward"
        TargetGroupArn: !Ref myTargetGroup
    SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
    Certificates:
      - CertificateArn: new-default-certificate-arn
```

將憑證新增至憑證清單

您可以使用以下程序，將憑證新增至接聽程式的憑證清單。如果您使用 建立接聽程式 AWS 管理主控台，我們會將預設憑證新增至憑證清單。否則，憑證清單為空白。將預設憑證新增至憑證清單可確保此憑證會與 SNI 通訊協定搭配使用，即使它被取代為預設憑證。如需詳細資訊，請參閱Application Load Balancer 的 SSL 憑證。

Console

將憑證新增至憑證清單

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選擇通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 選擇 Certificates (憑證) 索引標籤。
6. 若要將預設憑證新增至清單，請選擇新增預設憑證至清單。
7. 若要將非預設憑證新增至清單，請執行下列動作：
   a. 選擇新增憑證。
   b. 若要新增已由 ACM 或 IAM 管理的憑證，請選取憑證的核取方塊，然後選擇 Include as pending below (將以下列入待辦事項)。
   c. 若要新增不是由 ACM 或 IAM 管理的憑證，請選擇匯入憑證，完成表單，然後選擇匯入。
   d. 選擇新增待定憑證。

AWS CLI

將憑證新增至憑證清單

使用 add-listener-certificates 命令。

```
aws elbv2 add-listener-certificates \
    --listener-arn listener-arn \
    --certificates \
        CertificateArn=certificate-arn-1 \
        CertificateArn=certificate-arn-2 \
        CertificateArn=certificate-arn-3
```

CloudFormation

將憑證新增至憑證清單

定義 AWS::ElasticLoadBalancingV2::ListenerCertificate 類型的資源。

```
Resources:
  myCertificateList:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerCertificate'
    Properties:
      ListenerArn: !Ref myTLSListener
      Certificates:
        - CertificateArn: "certificate-arn-1"
        - CertificateArn: "certificate-arn-2"
        - CertificateArn: "certificate-arn-3"
```

從憑證清單中移除憑證

您可以使用以下程序，從 HTTPS 接聽程式的憑證清單中移除憑證。移除憑證之後，接聽程式就無法再使用該憑證建立連線。為了確保用戶端不受影響，請將新憑證新增至清單，並確認連線正在運作中，再從清單中移除憑證。

若要移除 TLS 接聽程式的預設憑證，請參閱更換預設憑證。

Console

從憑證清單中移除憑證

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。

5. 在憑證索引標籤上，選取憑證的核取方塊，然後選擇移除。
6. 出現確認提示時，請輸入 **confirm**，然後選擇移除。

AWS CLI

從憑證清單中移除憑證

使用 remove-listener-certificates 命令。

```
aws elbv2 remove-listener-certificates \
    --listener-arn listener-arn \
    --certificates CertificateArn=certificate-arn
```

更新安全政策

建立 HTTPS 接聽程式時，您可以選取符合您的需求的安全政策。新增安全政策後，您可以更新 HTTPS 接聽程式，以使用新的安全政策。Application Load Balancers 不支援自訂安全政策。如需詳細資訊，請參閱Application Load Balancer 的安全政策。

如果負載平衡器正在處理大量流量，則更新安全政策可能會導致中斷。若要降低負載平衡器處理大量流量時中斷的可能性，請建立額外的負載平衡器以協助處理流量或請求 LCU 保留。

相容性

- 連接到相同負載平衡器的所有安全接聽程式都必須使用相容的安全政策。若要將所有負載平衡器的安全接聽程式遷移至與目前正在使用的安全政策不相容的安全政策，請移除其中一個安全接聽程式以外的所有安全接聽程式、變更安全接聽程式的安全政策，然後建立其他安全接聽程式。
  - FIPS 後量子 TLS 政策和 FIPS 政策 - 相容
  - 後量子 TLS 政策和 FIPS 或 FIPS 後量子 TLS 政策 - 相容
  - TLS 政策（非 FIPS、non-post-quantum) 和 FIPS 或 FIPS 後量子 TLS 政策 - 不相容
  - TLS 政策（非 FIPS、non-post-quantum) 和後量子 TLS 政策 - 不相容

Console

更新安全政策

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 在安全索引標籤上，選擇編輯安全接聽程式設定。
6. 在安全接聽程式設定區段的安全政策下，選擇新的安全政策。
7. 選擇儲存變更。

AWS CLI

更新安全政策

使用 modify-listener 命令。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --ssl-policy ELBSecurityPolicy-TLS13-1-2-Res-2021-06
```

CloudFormation

更新安全政策

使用新的安全政策更新 AWS::ElasticLoadBalancingV2::Listener 資源。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
      Certificates:
        - CertificateArn: certificate-arn
```

HTTP 標頭修改

HTTP 標頭修改可讓您重新命名特定負載平衡器產生的標頭、插入特定回應標頭，以及停用伺服器回應標頭。Application Load Balancer 支援對請求和回應標頭進行標頭修改。

如需詳細資訊，請參閱為您的 Application Load Balancer 啟用 HTTP 標頭修改。

Application Load Balancer 的接聽程式規則

Application Load Balancer 的接聽程式規則會決定如何將請求路由到目標。當接聽程式收到請求時，它會依優先順序針對每個規則評估請求，從編號最低的規則開始。每個規則都包含要符合的條件，以及在符合規則的條件時要執行的動作。這種靈活的路由機制可讓您實作複雜的流量分佈模式，支援單一負載平衡器後方的多個應用程式或微服務，並根據應用程式的特定需求自訂請求處理。

規則基本概念

- 每個規則都包含下列元件：優先順序、動作、條件和選用轉換。
- 每個規則動作都有一個類型和執行動作所需的資訊。
- 每個規則條件都有一個類型和評估條件所需的資訊。
- 每個規則轉換都有要比對的規則運算式和替換字串。
- 規則條件和規則轉換中使用的規則表達式不支援下列功能： lookaheads、lookbehinds、backreferences、atomic group、hayive quantifiers、subroutines、recursion 和 Unicode 字元類別 （ 例如 `\p{L}`)。
- 建立接聽程式時，您會定義預設規則的預設動作。預設規則不能有條件或轉換。如果不符合任何其他規則的任何條件，則會執行預設規則的動作。
- 依優先順序評估規則，從最低值到最高值。預設規則最後評估。您無法變更預設規則的優先順序。
- 每個規則必須包含剛好以下其中一個動作：`forward`、`redirect` 或 `fixed-response`，而且必須是最後要執行的動作。
- 預設規則以外的每個規則都可以選擇性地包含下列其中一個條件：`host-header`、`path-pattern`、 `http-request-method`和 `source-ip`。它也可以選擇性地包含下列其中一個或兩個條件： `http-header`和 `query-string`。
- 預設規則以外的每個規則都可以選擇性地包含一個主機標頭重寫轉換和一個 URL 重寫轉換。
- 每個條件最多可指定三個比較字串，每個規則最多可指定五個比較字串。

目錄

- 接聽程式規則的動作類型
- 接聽程式規則的條件類型
- 接聽程式規則的轉換
- 為您的 Application Load Balancer 新增接聽程式規則
- 編輯 Application Load Balancer 的接聽程式規則
- 刪除 Application Load Balancer 的接聽程式規則

接聽程式規則的動作類型

動作會決定當接聽程式規則的條件滿足時，負載平衡器如何處理請求。每個規則必須至少有一個動作，指定如何處理相符的請求。每個規則動作都有類型和組態資訊。Application Load Balancer 支援接聽程式規則的下列動作類型。

動作類型

`authenticate-cognito`

[HTTPS 接聽程式] 使用 Amazon Cognito 來驗證使用者身分。如需詳細資訊，請參閱使用者身分驗證。

`authenticate-oidc`

[HTTPS 接聽程式] 使用與 OpenID Connect (OIDC) 相容的身分提供者來驗證使用者。如需詳細資訊，請參閱使用者身分驗證。

`fixed-response`

傳回自訂的 HTTP 回應。如需詳細資訊，請參閱固定回應動作。

`forward`

將請求轉送到指定的目標群組。如需詳細資訊，請參閱轉送動作。

`jwt-validation`

驗證用戶端請求中的 JWT 存取字符。如需詳細資訊，請參閱JWT 驗證。

`redirect`

將請求從一個 URL 重新導向到另一個 URL。如需詳細資訊，請參閱重新導向動作。

動作基本概念

- 每個規則必須僅包含下列其中一個路由動作：forward、 redirect或 fixed-response，而且必須是要執行的最後一個動作。
- HTTPS 接聽程式可以具有具有使用者身分驗證動作和路由動作的規則。
- 有多個動作時，會先執行優先順序最低的動作。
- 如果通訊協定版本為 gRPC 或 HTTP/2，則唯一支援的動作是 forward 動作。

固定回應動作

fixed-response 動作會捨棄用戶端請求並傳回自訂 HTTP 回應。您可以使用此動作來傳回 2XX、4XX 或 5XX 回應代碼和選用的訊息。

採取 fixed-response 動作時，會在存取日誌中記錄重新導向目標的動作和 URL。如需詳細資訊，請參閱存取日誌項目。會在 HTTP_Fixed_Response_Count 指標中報告成功的 fixed-response 動作計數。如需詳細資訊，請參閱Application Load Balancer 指標。

Example固定回應動作範例

您可以在建立或修改規則時指定動作。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。下列動作傳送包含指定的狀態碼和訊息本文的固定回應。

```
[
  {
      "Type": "fixed-response",
      "FixedResponseConfig": {
          "StatusCode": "200",
          "ContentType": "text/plain",
          "MessageBody": "Hello world"
      }
  }
]
```

轉送動作

forward 動作會將請求路由至它的目標群組。新增 forward 動作之前，請建立目標群組並將目標新增至群組。如需詳細資訊，請參閱建立目標群組。

將流量分配至多個目標群組

如果您為一個 forward 動作指定多個目標群組，則必須為每個目標群組指定加權。每個目標群組權重為介於 0 到 999 之間的值。符合加權目標群組之監聽程式規則的請求，會根據其權重分配到這些目標群組。例如，如果您指定兩個目標群組，每個目標群組的權重為 10，則每個目標群組都會收到一半的請求。如果您指定兩個目標群組，一個權重為 10，另一個權重為 20，則權重為 20 的目標群組接收的請求數量是另一個目標群組的兩倍。

如果您設定規則在加權目標群組和其中一個目標群組之間分配流量是空的，或只有運作狀態不佳的目標，負載平衡器不會自動容錯移轉至具有運作狀態良好目標的目標群組。

黏性工作階段和加權目標群組

根據預設，在加權的目標群組之間分配流量設定規則，並不保證可以接受黏性工作階段。若要確保接受黏性工作階段，請啟用目標群組的黏性規則。當負載平衡器首先將請求路由至加權目標群組時，它會產生名為 AWSALBTG 的 cookie，以編碼選取的目標群組相關資訊、加密 cookie，並在回應用戶端時包含 cookie。用戶端應該包含負載平衡器後續請求中接收的 cookie。當負載平衡器收到符合已啟用目標群組粘性的規則的請求並包含 cookie 時，會將請求路由至 cookie 中指定的目標群組。

Application Load Balancer 不支援 URL 編碼的 Cookie 值。

透過 CORS (跨來源資源共享) 請求，有些瀏覽器需要 SameSite=None; Secure 來啟用綁定。在此案例中，Elastic Load Balancing 會產生第二個 Cookie (AWSALBTGCORS)，其中包含與原始粘性 Cookie 相同的資訊以及此 SameSite 屬性。用戶端會同時收到這兩個 Cookie。

具有一個目標群組的轉送動作範例

您可以在建立或修改規則時指定動作。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。下列動作將請求轉送到指定的目標群組。

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067"
              }
          ]
      }
  }
]
```

具有加權目標群組的向前動作範例

下列動作會根據每個目標群組的權重，將要求轉送至兩個指定的目標群組。

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
                  "Weight": 10
              },
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
                  "Weight": 20
              }
          ]
      }
  }
]
```

已啟用粘性的轉送動作範例

如果您有一個轉送動作涉及多個目標群組，且一個或多個目標群組已啟用粘性會話，則您必須啟用目標群組粘性。

下列動作會將請求轉送至兩個指定的目標群組，搭配啟用目標群組黏性。不包含該綁定 Cookie 的請求會根據每個目標群組的權重進行路由。

```
[
  {
      "Type": "forward",
      "ForwardConfig": {
          "TargetGroups": [
              {
                  "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
                  "Weight": 10
              },
              {
```

```
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
                    "Weight": 20
                }
            ],
            "TargetGroupStickinessConfig": {
                "Enabled": true,
                "DurationSeconds": 1000
            }
        }
    }
]
```

重新導向動作

`redirect` 動作會將用戶端請求從一個 URL 重新導向到另一個 URL。您可以根據您的需求，將重新導向設定為暫時 (HTTP 302) 或永久 (HTTP 301)。

URI 包含以下元件：

```
protocol://hostname:port/path?query
```

您必須修改以下至少一個元件，以避免重新導向迴圈：protocol、hostname、port 或 path。未修改的任何元件會維持其原始值。

protocol

通訊協定 (HTTP 或 HTTPS)。您可以重新導向 HTTP 到 HTTP、HTTP 到 HTTPS，以及 HTTPS 到 HTTPS。您不能重新導向 HTTPS 到 HTTP。

hostname

主機名稱。主機名稱不區分大小寫、長度最多 128 個字元，由英數字元、萬用字元 (* 和 ?) 和連字號 (-) 組成。

port

連接埠 (1 到 65535)。

路徑

絕對路徑，開頭為前置字元 "/"。路徑區分大小寫、長度最多 128 個字元，由英數字元、萬用字元 (* 和 ?)、& (使用 &) 和下列特殊字元組成：_-.$/~'"@:+。

query

查詢參數。長度上限為 128 個字元。

您可以使用以下預留關鍵字，來在目標 URL 中重複使用原始 URL 的 URI 元件：

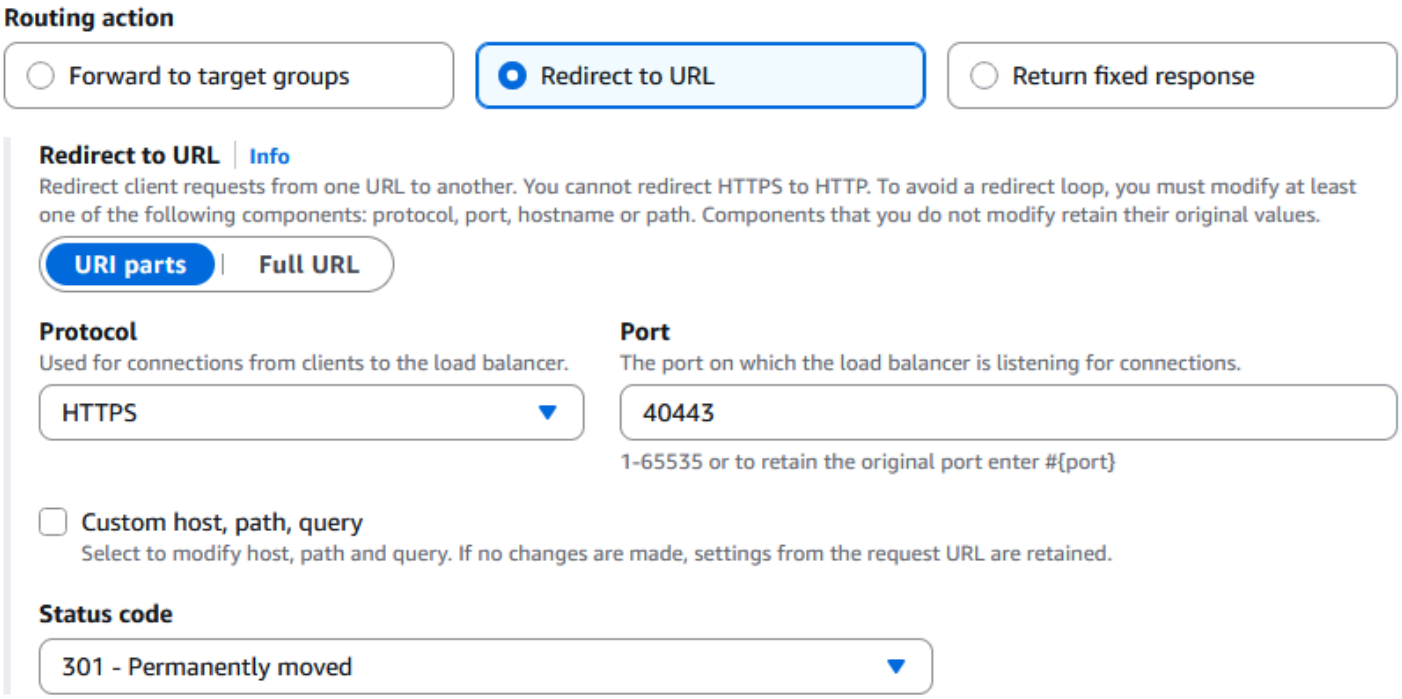
- `#{protocol}` - 保留通訊協定。用於通訊協定和查詢元件。
- `#{host}` - 保留網域。用於主機名稱、路徑和查詢元件。
- `#{port}` - 保留連接埠。用於連接埠、路徑和查詢元件。
- `#{path}` - 保留路徑。用於路徑和查詢元件。
- `#{query}` - 保留查詢參數。用於查詢元件。

採取 `redirect` 動作時，會將動作記錄於存取日誌中。如需詳細資訊，請參閱存取日誌項目。會在 `HTTP_Redirect_Count` 指標中報告成功的 `redirect` 動作計數。如需詳細資訊，請參閱Application Load Balancer 指標。

使用主控台的重新導向動作範例

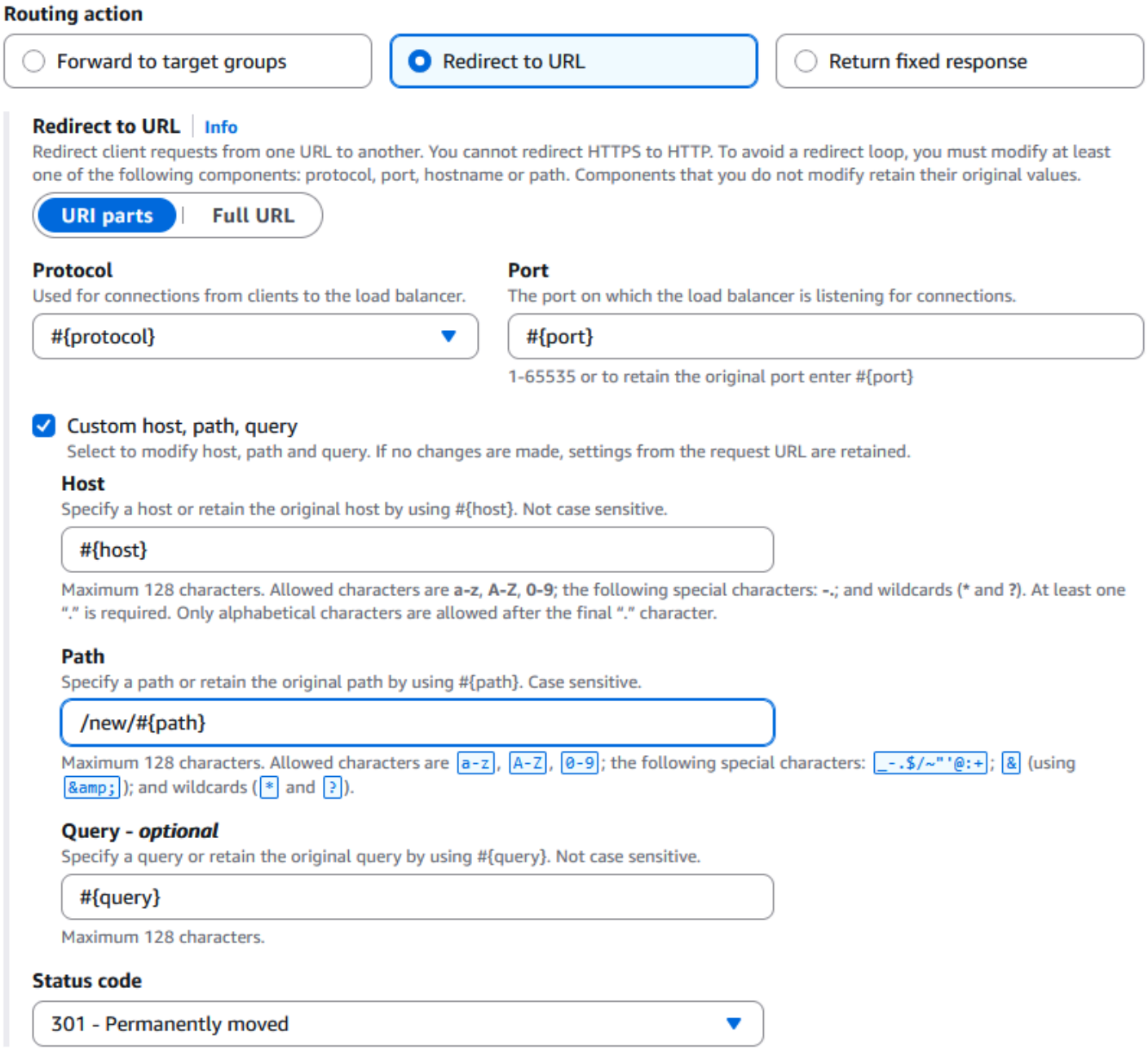
使用 HTTPS 和連接埠 40443 重新導向

以下規則會設定使用 HTTPS 通訊協定和指定連接埠 (40443) 永久重新導向到 URL，但會保留原始主機名稱、路徑和查詢參數。這個畫面等同於 "https://#{host}:40443/#{path}?#{query}"。



使用修改過的路徑重新導向

以下規則會設定永久重新導向到 URL，保留通訊協定、連接埠、主機名稱和查詢參數，並使用 `#{path}` 關鍵字來建立修改的路徑。這個畫面等同於 "#{protocol}://#{host}:#{port}/new/#{path}?#{query}"。



使用 重新導向動作的範例 AWS CLI

使用 HTTPS 和連接埠 40443 重新導向

您可以在建立或修改規則時指定動作。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。下列動作將 HTTP 請求重新導向到連接埠 443 的 HTTPS 請求，而且使用與 HTTP 請求相同的主機名稱、路徑和查詢字串。

```
--actions '[{
    "Type": "redirect",
    "RedirectConfig": {
        "Protocol": "HTTPS",
        "Port": "443",
        "Host": "#{host}",
        "Path": "/#{path}",
        "Query": "#{query}",
        "StatusCode": "HTTP_301"
    }
}]'
```

接聽程式規則的條件類型

條件定義傳入請求必須符合的條件，接聽程式規則才能生效。如果請求符合規則的條件，則會依規則的動作指定來處理請求。每個規則條件具有類型和組態資訊。Application Load Balancer 支援接聽程式規則的下列條件類型。

條件類型

`host-header`

根據每個請求的主機名稱來路由傳送。如需詳細資訊，請參閱主機條件。

`http-header`

根據每個請求的 HTTP 標頭來路由傳送。如需詳細資訊，請參閱HTTP 標頭條件。

`http-request-method`

根據每個請求的 HTTP 請求方法來路由傳送。如需詳細資訊，請參閱HTTP 請求方法條件。

`path-pattern`

根據請求 URL 中的路徑模式來路由傳送。如需詳細資訊，請參閱路徑條件。

`query-string`

根據查詢字串中的鍵值組或值來路由傳送。如需詳細資訊，請參閱查詢字串條件。

`source-ip`

根據每個請求的來源 IP 位址來路由傳送。如需詳細資訊，請參閱來源 IP 地址條件。

條件基本概念

- 每個規則都可以選擇性地包含下列每個條件的零或其中一個：`host-header`、`path-pattern`、`http-request-method`和 `source-ip`。每個規則也可以包含下列每個條件的零或多個： `http-header`和 `query-string`。
- 透過 `host-header`、 `http-header`和 `path-pattern`條件，您可以使用值比對或規則表達式 (regex) 比對。
- 每個條件最多可以指定三個比對評估。例如，對於每個 `http-header` 條件，您最多可以指定三個字串，以便與請求中的 HTTP 標頭值做比較。如果其中一個字串符合 HTTP 標頭的值，即符合條件。若要求所有字串都要符合，請為每個比對評估建立一個條件。
- 每個規則最多可以指定五個比對評估。例如，您可以建立含有五個條件的規則，其中每個條件有一個比對評估。
- 您可以在 `http-header`、`host-header`、`path-pattern` 和 `query-string` 條件的比對評估中包含萬用字元。每個規則以五個萬用字元為限。
- 規則僅會套用至可見的 ASCII 字元；會排除控制字元 (0x00 到 0x1f 和 0x7f)。
- 規則條件中使用的規則表達式不支援下列功能： lookaheads、 lookbehinds、backreferences、atomic group、haative quantifiers、subroutines、recursion 和 Unicode 字元類別 （ 例如 `\p{L}`)。

示範

如需示範，請參閱 Advanced Request Routing。

主機條件

您可以使用主機條件來定義規則，以根據主機標頭中的主機名稱來路由傳送請求 (也稱為以主機為基礎的路由)。這可讓您使用單一負載平衡器來支援多個子網域和不同的頂層網域。

主機名稱不區分大小寫，長度最多可達 128 個字元，而且可以包含下列任何字元：

- A-Z、a-z、0-9
- - .
- * (符合 0 個或多個字元)

- ? (確切符合 1 個字元)

您必須至少包含一個 "." 字元。您只可以在最後的 "." 字元之後包含字母字元。

主機名稱範例

- example.com
- test.example.com
- *.example.com

規則 *.example.com 會符合 test.example.com 但不會符合 example.com。

Example主機標頭條件範例

您可以在建立或修改規則時指定條件。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。

Value matching

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "Values": ["*.example.com"]
      }
  }
]
```

Regex matching

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "RegexValues": ["^(.*)\\.example\\.com$"]
      }
  }
]
```

HTTP 標頭條件

您可以使用 HTTP 標頭條件來設定規則，以根據請求的 HTTP 標頭來路由傳送請求。您可以指定標準或自訂 HTTP 標頭欄位的名稱。標頭名稱和比對評估不區分大小寫。比較字串中支援下列萬用字元：* (符合 0 個或多個字元) 和 ？(確切符合 1 個字元)。標頭名稱中不支援萬用字元。

`routing.http.drop_invalid_header_fields` 啟用 Application Load Balancer 屬性時，它會捨棄不符合規則表達式 () 的標頭名稱`A-Z,a-z,0-9`。也可以新增不符合規則表達式的標頭名稱。

Example HTTP 標頭條件範例

您可以在建立或修改規則時指定條件。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。當請求的 User-Agent 標頭符合其中一個指定的字串時，即滿足下列條件。

Value matching

```
[
  {
      "Field": "http-header",
      "HttpHeaderConfig": {
          "HttpHeaderName": "User-Agent",
          "Values": ["*Chrome*", "*Safari*"]
      }
  }
]
```

Regex matching

```
[
  {
      "Field": "http-header",
      "HttpHeaderConfig": {
          "HttpHeaderName": "User-Agent",
          "RegexValues": [".+"]
      }
  }
]
```

HTTP 請求方法條件

您可以使用 HTTP 請求方法條件來設定規則，以根據請求的 HTTP 請求方法來路由傳送請求。您可以指定標準或自訂 HTTP 方法。比對評估區分大小寫。不支援萬用字元；因此，方法名稱必須完全相符。

建議您以相同方式來路由傳送 GET 和 HEAD 請求，因為可快取對 HEAD 請求的回應。

Example HTTP 方法條件範例

您可以在建立或修改規則時指定條件。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。使用指定方法的請求符合下列條件。

```
[
  {
      "Field": "http-request-method",
      "HttpRequestMethodConfig": {
          "Values": ["CUSTOM-METHOD"]
      }
  }
]
```

路徑條件

您可以使用路徑條件來定義規則，以根據請求中的 URL 來路由傳送請求 (也稱為以路徑為基礎的路由)。

系統只會將路徑模式套用到 URL 的路徑，而不會套用到其查詢參數。它僅適用於可見的 ASCII 字符；會排除控制字符 (0x00 到 0x1f 和 0x7f)。

只有在 URI 標準化發生後，才會執行規則評估。

名稱模式區分大小寫，長度最多可達 128 個字元，而且可以包含下列任何字元。

- A-Z、a-z、0-9
- _ - . $ / ~ " ' @ : +
- & (使用 &)
- * (符合 0 個或多個字元)
- ? (確切符合 1 個字元)

如果通訊協定版本為 gRPC，則條件可以具體到套件、服務或方法。

HTTP 路徑模式範例

- `/img/*`
- `/img/*/pics`

gRPC 路徑模式範例

- /package
- /package.service
- /package.service/method

路徑模式是用於路由請求，但不會修改請求。例如，如果規則具有 `/img/*` 的路徑模式，則該規則會將 `/img/picture.jpg` 的請求轉送到指定的目標群組，作為對 `/img/picture.jpg` 的請求。

Example路徑模式條件範例

您可以在建立或修改規則時指定條件。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。當請求的 URL 包含指定的字串時，即滿足下列條件。

Value matching

```
[
  {
      "Field": "path-pattern",
      "PathPatternConfig": {
          "Values": ["/img/*"]
      }
  }
]
```

Regex matching

```
[
  {
      "Field": "path-pattern",
      "PathPatternConfig": {
          "RegexValues": ["^\\/api\\/(.*)$"]
      }
```

```
    }
  ]
```

查詢字串條件

您可以使用查詢字串條件來設定規則，以根據查詢字串中的鍵值組或值來路由傳送請求。比對評估不區分大小寫。支援下列萬用字元：* (符合 0 個或多個字元) 和 ? (確切符合 1 個字元)。

Example查詢字串條件範例

您可以在建立或修改規則時指定條件。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。當請求的查詢字串包含鍵值組 "version=v1" 或任何設為 "example" 的索引鍵時，即滿足下列條件。

```
[
  {
      "Field": "query-string",
      "QueryStringConfig": {
          "Values": [
            {
                "Key": "version",
                "Value": "v1"
            },
            {
                "Value": "*example*"
            }
          ]
      }
  }
]
```

來源 IP 地址條件

您可以使用來源 IP 地址條件來設定規則，以根據請求的來源 IP 地址來路由傳送請求。必須以 CIDR 格式指定 IP 地址。IPv4 和 IPv6 地址都可用。不支援萬用字元。您無法指定來源 IP 規則條件的 `255.255.255.255/32` CIDR。

如果用戶端位在 proxy 後方，則此為 proxy 的 IP 地址，而不是用戶端的 IP 地址。

X-Forwarded-For 標頭中的地址不滿足此條件。若要搜尋 X-Forwarded-For 標頭中的地址，請使用 `http-header` 條件。

Example來源 IP 條件範例

您可以在建立或修改規則時指定條件。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。當請求的來源 IP 地址出現在其中一個指定的 CIDR 區塊時，即滿足下列條件。

```
[
  {
      "Field": "source-ip",
      "SourceIpConfig": {
          "Values": ["192.0.2.0/24", "198.51.100.10/32"]
      }
  }
]
```

接聽程式規則的轉換

規則轉換會在傳入請求路由至目標之前將其重寫。重寫請求不會變更評估規則條件時所做的路由決策。當用戶端傳送不同於目標預期的 URL 或主機標頭時，這很有用。

使用規則轉換會將修改路徑、查詢字串和主機標頭的責任卸載至負載平衡器。這不需要將自訂修改邏輯新增至您的應用程式程式碼，或依賴第三方代理來執行修改。

Application Load Balancer 支援接聽程式規則的下列轉換。

轉換

`host-header-rewrite`

重寫請求中的主機標頭。轉換會使用規則表達式來比對主機標頭中的模式，然後將其取代為替代字串。

`url-rewrite`

重寫請求 URL。轉換會使用規則表達式來比對請求 URL 中的模式，然後將其取代為替代字串。

轉換基本概念

- 您可以為每個規則新增一個主機標頭重寫轉換和一個 URL 重寫轉換。
- 您無法將轉換新增至預設規則。
- 如果沒有模式相符項目，則會將原始請求傳送至目標。

- 如果模式相符但轉換失敗，我們會傳回 HTTP 500 錯誤。
- 規則轉換中使用的規則表達式不支援下列功能： lookaheads、 lookbehinds、backreferences、atomic group、haative quantifiers、subroutines、recursion 和 Unicode 字元類別（例如 `\p{L}`)。

主機標頭重寫轉換

您可以修改主機標頭中指定的網域名稱。

Example主機標頭轉換範例

您可以在建立或修改規則時指定轉換。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。以下是範例主機標頭轉換。它會將主機標頭轉換為內部端點。

```
[
  {
      "Type": "host-header-rewrite",
      "HostHeaderRewriteConfig": {
          "Rewrites": [
              {
                  "Regex": "^mywebsite-(.+).com$",
                  "Replace": "internal.dev.$1.myweb.com"
              }
          ]
      }
  }
]
```

例如，此轉換會將主機標頭重寫`https://mywebsite-example.com/project-a`為 `https://internal.dev.example.myweb.com/project-a`。

URL 重寫轉換

您可以修改 URL 的路徑或查詢字串。透過在負載平衡器層級重寫 URL，即使後端服務變更，使用者和搜尋引擎的前端 URLs也可以保持一致。您也可以簡化複雜的 URL 查詢字串，讓客戶更容易輸入。

請注意，您無法修改 URL 的通訊協定或連接埠，只能修改路徑和查詢字串。

Example範例 URL 重寫轉換

您可以在建立或修改規則時指定轉換。如需詳細資訊，請參閱 create-rule 和 modify-rule 命令。以下是 URL 重寫轉換的範例。它會將目錄結構轉換為查詢字串。

```
[
  {
      "Type": "url-rewrite",
      "UrlRewriteConfig": {
          "Rewrites": [
              {
                  "Regex": "^/dp/([A-Za-z0-9]+)/?$",
                  "Replace": "/product.php?id=$1"
              }
          ]
      }
  }
]
```

例如，此轉換會將請求 URL 重寫`https://www.example.com/dp/B09G3HRMW`為 `https://www.example.com/product.php?id=B09G3HRMW`。

URL 重寫與 URL 重新導向的差異

特性	URL 重新導向	URL 重寫
URL 顯示	瀏覽器地址列中的變更	瀏覽器地址列沒有變更
狀態碼	使用 301 （永久）或 302 （暫時）	無狀態碼變更
處理	瀏覽器端	伺服器端
常見用途	網域變更、網站合併、修正中斷的連結	清除 SEO URLs、隱藏複雜的結構、提供舊版 URL 映射

為您的 Application Load Balancer 新增接聽程式規則

您可以在建立接聽程式時定義預設規則。您可以隨時定義其他規則。每個規則都必須指定動作和條件，並且可以選擇指定轉換。如需詳細資訊，請參閱下列內容：

- 動作類型
- 條件類型
- 轉換

Console

新增規則

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 在規則索引標籤上選擇新增規則。
6. （選用）若要指定規則的名稱，請展開名稱和標籤，然後輸入名稱。若要新增其他標籤，請選擇新增其他標籤，然後輸入標籤索引鍵和標籤值。
7. 針對每個條件，選擇新增條件，選擇條件類型，並提供必要的條件值：

   - 主機標頭 – 選取相符模式類型，然後輸入主機標頭。

     值比對 – 最多 128 個字元。不區分大小寫。允許的字元有 a-z、A-Z、0-9，以及下列特殊字元：-_.; 和萬用字元 (* 和 ?)。您必須至少包含一個 "." 字元。您只可以在最後的 "." 字元之後包含字母字元。

     Regex 比對 – 最多 128 個字元。
   - 路徑 – 選取比對模式類型，然後輸入路徑。

     值比對 – 最多 128 個字元。區分大小寫。允許的字元有 a-z、A-Z、0-9，以及下列特殊字元：_-.$/~'"@:+; &; 和萬用字元 (* 和 ?)。

     Regex 比對 – 最多 128 個字元。
   - 查詢字串 – 輸入 key：value 對，或不含 key 的值。

     最多 128 個字元。不區分大小寫。允許的字元有 a-z、A-Z、0-9，以及下列特殊字元： _-.$/~'"@:+&()!,;=; 和萬用字元 (* 和 ?)。
   - HTTP 請求方法 – 輸入 HTTP 請求方法。

     最多 40 個字元。區分大小寫。允許的字元有 A-Z 以及下列特殊字元：-_。不支援萬用字元。
   - HTTP 標頭 – 選取比對模式類型，然後輸入標頭的名稱和比較字串。
     - HTTP 標頭名稱 – 規則會評估內含此標頭的請求，以確認相符值。

值比對 – 最多 40 個字元。不區分大小寫。允許的字元有 a-z、A-Z、0-9 以及下列特殊字元：*?-!#$%&'+.^_`|~。不支援萬用字元。

Regex 比對 – 最多 128 個字元。

- HTTP 標頭值 – 輸入要與 HTTP 標頭值比較的字串。

  值相符 最多 128 個字元。不區分大小寫。允許的字元為 a-z、A-Z、0-9； 空格；下列特殊字元：！"#$%&'()+,./:;<=>@【】^_`{|}~-； 和萬用字元 (* 和 ？)。

  Regex 比對 – 最多 128 個字元。

- 來源 IP – 以 CIDR 格式定義來源 IP 地址。IPv4 和 IPv6 CIDR 都是允許的。不支援萬用字元。

8. （選用）若要新增轉換，請選擇新增轉換、選擇轉換類型，然後輸入要比對的規則運算式和替代字串。
9. （選用，僅限 HTTPS 接聽程式）對於預先路由動作，選取下列其中一個動作：
   - 驗證使用者 – 選擇身分提供者並提供必要的資訊。如需詳細資訊，請參閱使用 Application Load Balancer 來驗證使用者身分。
   - 驗證字符 – 輸入 JWKS 端點、問題和任何其他宣告。如需詳細資訊，請參閱使用 Application Load Balancer 驗證 JWTs。
10. 針對路由動作，選取下列其中一個動作：
    - 轉送至目標群組 – 選擇目標群組。若要新增另一個目標群組，請選擇新增目標群組、選擇目標群組、檢閱相對權重，並視需要更新權重。如果您在任何目標群組上啟用粘性，則必須啟用群組層級粘性。
    - 重新導向至 URL – 在 URI 部分索引標籤上分別輸入每個部分，或在完整 URL 索引標籤上輸入完整地址，以輸入 URL。針對狀態碼，根據您的需求選取暫時 (HTTP 302) 或永久 (HTTP 301)。
    - 傳回固定回應 – 輸入回應碼以針對捨棄的用戶端請求傳回。或者，您可以指定內容類型和回應內文。
11. 選擇下一步。
12. 針對 Priority，輸入 1-50，000 的值。規則會以從最低值到最高值的優先順序進行評估。
13. 選擇下一步。
14. 在 Review and create (檢閱和建立) 頁面上，選取 Create (建立)。

AWS CLI

新增規則

使用 create-rule 命令。

下列範例會建立具有 forward動作和 host-header條件的規則。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
    --priority 10 \
    --conditions "Field=host-header,Values=example.com,www.example.com" \
    --actions "Type=forward,TargetGroupArn=target-group-arn"
```

若要建立在兩個目標群組之間分配流量的轉送動作，請改用下列--actions選項。

```
    --actions '[{
        "Type":"forward",
        "ForwardConfig":{
          "TargetGroups":[
            {"TargetGroupArn":"target-group-1-arn","Weight":50},
            {"TargetGroupArn":"target-group-2-arn","Weight":50}
          ]
        }
    }]'
```

下列範例會建立具有 fixed-response動作和 source-ip條件的規則。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
    --priority 20 \
    --conditions '[{"Field":"source-ip","SourceIpConfig":{"Values":
["192.168.1.0/24","10.0.0.0/16"]}}]' \
    --actions "Type=fixed-
response,FixedResponseConfig={StatusCode=403,ContentType=text/
plain,MessageBody='Access denied'}"
```

下列範例會建立具有 redirect動作和 http-header條件的規則。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
```

```
    --priority 30 \
    --conditions '[{"Field":"http-header","HttpHeaderConfig":
{"HttpHeaderName":"User-Agent","Values":["*Mobile*","*Android*","*iPhone*"]}}]' \
    --actions
 "Type=redirect,RedirectConfig={Host=m.example.com,StatusCode=HTTP_302}"
```

CloudFormation

新增規則

定義 AWS::ElasticLoadBalancingV2::ListenerRule 類型的資源。

下列範例會建立具有 forward動作和 host-header條件的規則。符合條件時，規則會將流量傳送至指定的目標群組。

```
Resources:
    myForwardListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 10
       Conditions:
         - Field: host-header
           Values:
             - example.com
             - www.example.com
       Actions:
         - Type: forward
           TargetGroupArn: !Ref myTargetGroup
```

或者，若要建立轉送動作，在滿足條件時在兩個目標群組之間分配流量，請定義 Actions ，如下所示。

```
       Actions:
         - Type: forward
           ForwardConfig:
             TargetGroups:
               - TargetGroupArn: !Ref TargetGroup1
                 Weight: 50
               - TargetGroupArn: !Ref TargetGroup2
                 Weight: 50
```

下列範例會建立具有 fixed-response動作和 source-ip條件的規則。

```
Resources:
    myFixedResponseListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 20
       Conditions:
         - Field: source-ip
           SourceIpConfig:
             Values:
                - 192.168.1.0/24
                - 10.0.0.0/16
       Actions:
         - Type: fixed-response
           FixedResponseConfig:
             StatusCode: 403
             ContentType: text/plain
             MessageBody: "Access denied"
```

下列範例會建立具有 redirect動作和 http-header條件的規則。

```
Resources:
    myRedirectListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 30
       Conditions:
         - Field: http-header
           HttpHeaderConfig:
             HttpHeaderName: User-Agent
             Values:
               - "*Mobile*"
               - "*Android*"
               - "*iPhone*"
       Actions:
         - Type: redirect
           RedirectConfig:
             Host: m.example.com
             StatusCode: HTTP_302
```

編輯 Application Load Balancer 的接聽程式規則

您可以隨時編輯接聽程式規則的動作和條件。規則更新不會立即生效，因此在您更新規則後，可以使用先前的規則組態路由傳送請求一段時間。任何進行中的請求都已完成。

任務

- 修改預設動作
- 更新規則優先順序
- 更新動作、條件和轉換
- 管理規則標籤

修改預設動作

預設動作會指派給名為 Default 的規則。您可以保留目前的規則類型並變更必要資訊，也可以變更規則類型並提供新的必要資訊。

Console

修改預設動作

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 在規則索引標籤的接聽程式規則區段中，選取預設規則。選擇動作、編輯規則。
6. 在預設動作下，視需要更新動作。

AWS CLI

修改預設動作

使用 modify-listener 命令。下列範例會更新 `forward`動作的目標群組。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --default-actions Type=forward,TargetGroupArn=new-target-group-arn
```

下列範例會更新預設動作，以在兩個目標群組之間平均分配流量。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --default-actions '[{
      "Type":"forward",
      "ForwardConfig":{
        "TargetGroups":[
          {"TargetGroupArn":"target-group-1-arn","Weight":50},
          {"TargetGroupArn":"target-group-2-arn","Weight":50}
        ]
      }
    }]'
```

CloudFormation

修改預設動作

更新 AWS::ElasticLoadBalancingV2::Listener 資源。

```
Resources:
    myHTTPlistener:
      Type: 'AWS::ElasticLoadBalancingV2::Listener'
      Properties:
        LoadBalancerArn: !Ref myLoadBalancer
        Protocol: HTTP
        Port: 80
        DefaultActions:
          - Type: "forward"
            TargetGroupArn: !Ref myNewTargetGroup
```

更新規則優先順序

依優先順序評估規則，從最低值到最高值。預設規則最後評估。您可以隨時變更非預設規則的優先順序。您無法變更預設規則的優先順序。

Console

更新規則優先順序

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 在規則索引標籤上，選取接聽程式規則，然後選擇動作、重新排定規則的優先順序。
6. 在接聽程式規則區段中，優先順序欄會顯示目前的規則優先順序。若要更新規則優先順序，請輸入 1-50，000 的值。
7. 選擇儲存變更。

AWS CLI

更新規則優先順序

使用 set-rule-priorities 命令。

```
aws elbv2 set-rule-priorities \
    --rule-priorities "RuleArn=listener-rule-arn,Priority=5"
```

CloudFormation

更新規則優先順序

更新 AWS::ElasticLoadBalancingV2::ListenerRule 資源。

```
Resources:
    myListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 5
       Conditions:
         - Field: host-header
           Values:
             - example.com
             - www.example.com
       Actions:
         - Type: forward
           TargetGroupArn: !Ref myTargetGroup
```

更新動作、條件和轉換

您可以更新規則的動作、條件和轉換。

Console

更新規則動作、條件和轉換

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 在規則索引標籤上，選取接聽程式規則，然後選擇動作、編輯規則。
6. 視需要更新動作、條件和轉換。如需詳細步驟，請參閱新增規則。
7. 選擇下一步。
8. （選用）更新優先順序。
9. 選擇下一步。
10. 選擇儲存變更。

AWS CLI

更新規則動作、條件和轉換

使用 modify-rule 命令。至少包含下列其中一個選項：`--actions`、 `--conditions`和 `--transforms`。

如需這些選項的範例，請參閱 新增規則。

CloudFormation

更新規則動作、條件和轉換

更新 AWS::ElasticLoadBalancingV2::ListenerRule 資源。

如需規則範例，請參閱 新增規則。

管理規則標籤

標籤可協助您以不同方式來分類接聽程式和規則。例如，您可以依用途、擁有者或環境來標記資源。每個規則的標籤索引鍵必須是唯一的。如果新增的標籤有已經和規則建立關聯的索引鍵，則此動作會更新該標籤的值。

當您使用完標籤之後，可以將其移除。

Console

管理規則的標籤

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選擇負載平衡器的名稱以開啟其詳細資訊頁面。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 在規則索引標籤上，選取名稱標籤欄中的文字，以開啟規則的詳細資訊頁面。
6. 在規則詳細資訊頁面上，選擇管理標籤。
7. 在管理標籤頁面上，可以執行下列一個或多個動作：
   a. 如要新增標籤，請選擇新增標籤，然後輸入索引鍵和值的值。
   b. 若要移除標籤，請選擇標籤旁的移除。
   c. 若要更新標籤，請輸入金鑰或值的新值。
8. 選擇儲存變更。

AWS CLI

將標籤新增至規則

使用 add-tags 命令。

```
aws elbv2 add-tags \
    --resource-arns listener-rule-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

從規則中移除標籤

使用 remove-tags 命令。

```
aws elbv2 remove-tags \
    --resource-arns listener-rule-arn \
    --tag-keys project department
```

CloudFormation

將標籤新增至規則

更新 AWS::ElasticLoadBalancingV2::ListenerRule 資源。

```
Resources:
    myListenerRule:
     Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
     Properties:
       ListenerArn: !Ref myListener
       Priority: 10
       Conditions:
         - Field: host-header
           Values:
             - example.com
             - www.example.com
       Actions:
         - Type: forward
           TargetGroupArn: !Ref myTargetGroup
       Tags:
        - Key: 'project'
          Value: 'lima'
        - Key: 'department'
          Value: 'digital-media'
```

刪除 Application Load Balancer 的接聽程式規則

您可以隨時刪除接聽程式的非預設規則。您無法刪除接聽程式的預設規則。刪除接聽程式時，其所有規則將被刪除。

Console

刪除規則

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取通訊協定：連接埠資料欄中的文字，以開啟接聽程式的詳細資訊頁面。
5. 選取規則。
6. 依序選擇 Actions (動作)、Delete rules (刪除規則)。
7. 出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

AWS CLI

刪除規則

使用 delete-rule 命令。

```
aws elbv2 delete-rule \
    --rule-arn listener-rule-arn
```

在 Application Load Balancer 中使用 TLS 進行相互身分驗證

相互 TLS 身分驗證是傳輸層安全性 (TLS) 的變化。傳統 TLS 會在伺服器和用戶端之間建立安全通訊，其中伺服器需要將其身分提供給其用戶端。使用交互 TLS，負載平衡器在交涉 TLS 時，在用戶端和伺服器之間交涉交互身分驗證。當您搭配 Application Load Balancer 使用交互 TLS 時，您可以簡化身分驗證管理並減少應用程式的負載。

透過使用交互 TLS，您的負載平衡器可以管理用戶端身分驗證，以協助確保只有信任的用戶端與您的後端應用程式通訊。當您使用此功能時，負載平衡器會使用來自第三方憑證授權單位 (CA) 的憑證，或選用 AWS 私有憑證授權單位 (PCA) 搭配撤銷檢查來驗證用戶端。負載平衡器會使用 HTTP 標頭將用戶端憑證資訊傳遞至後端，您的應用程式可用來進行授權。

Application Load Balancer 的相互 TLS 提供下列選項來驗證 X.509v3 用戶端憑證：

- 相互 TLS 傳遞：負載平衡器會將整個用戶端憑證鏈傳送至目標，而不進行驗證。目標應驗證用戶端憑證鏈。然後，使用用戶端憑證鏈，您可以在應用程式中實作負載平衡器身分驗證和目標授權邏輯。
- 相互 TLS 驗證：當負載平衡器交涉 TLS 連線時，負載平衡器會對用戶端執行 X.509 用戶端憑證驗證。

若要使用交互 TLS 傳遞，您必須設定接聽程式以接受來自用戶端的憑證。若要搭配驗證使用交互 TLS，請參閱 在 Application Load Balancer 上設定交互 TLS。

開始在 Application Load Balancer 上設定交互 TLS 之前

開始在 Application Load Balancer 上設定交互 TLS 之前，請注意下列事項：

配額

Application Load Balancer 包含與 AWS 帳戶中正在使用的信任存放區、CA 憑證和憑證撤銷清單數量相關的特定限制。

如需詳細資訊，請參閱 Quotas for your Application Load Balancers。

憑證的需求

Application Load Balancer 支援下列與交互 TLS 身分驗證搭配使用的憑證：

- 支援的憑證：X.509v3
- 支援的公有金鑰：RDA 2K – 8K 或 ECDSA secp256r1、secp384r1、secp521r1
- 支援的簽章演算法：SHA256、384、512 搭配 RSA/SHA256、384、512 搭配 EC/SHA256、384、512 雜湊搭配 RSASSA-PSS 搭配 MGF1

CA 憑證套件

下列適用於憑證授權機構 (CA) 套件：

- Application Load Balancer 會批次上傳每個憑證授權單位 (CA) 憑證套件。Application Load Balancer 不支援上傳個別憑證。如果您需要新增憑證，則必須上傳憑證套件檔案。
- 若要取代 CA 憑證套件，請使用 ModifyTrustStore API。

傳遞的憑證順序

當您使用交互 TLS 傳遞時，Application Load Balancer 會插入標頭，將用戶端憑證鏈呈現給後端目標。呈現順序從分葉憑證開始，並以根憑證結束。

工作階段恢復

搭配 Application Load Balancer 使用交互 TLS 傳遞或驗證模式時，不支援工作階段恢復。

HTTP 標頭

Application Load Balancer 使用 `X-Amzn-Mtls`標頭，在使用交互 TLS 交涉用戶端連線時傳送憑證資訊。如需詳細資訊和範例標頭，請參閱 HTTP 標頭和交互 TLS。

CA 憑證檔案

CA 憑證檔案必須符合下列要求：

- 憑證檔案必須使用 PEM （隱私權增強型郵件） 格式。
- 憑證內容必須括在 -----BEGIN CERTIFICATE-----和 -----END CERTIFICATE-----邊界內。
- 註解前面必須加上#字元，且不得包含任何-字元。
- 不能有任何空白行。

不接受的範例憑證 （無效）：

```
# comments

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 01
    Signature Algorithm: ecdsa-with-SHA384
        Issuer: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
        Validity
            Not Before: Jan 11 23:57:57 2024 GMT
            Not After : Jan 10 00:57:57 2029 GMT
        Subject: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (384 bit)
                pub:
                    00:01:02:03:04:05:06:07:08
                ASN1 OID: secp384r1
                NIST CURVE: P-384
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Subject Key Identifier:
                00:01:02:03:04:05:06:07:08
            X509v3 Subject Alternative Name:
                URI:EXAMPLE.COM
    Signature Algorithm: ecdsa-with-SHA384
         00:01:02:03:04:05:06:07:08
-----BEGIN CERTIFICATE-----
```

```
Base64-encoded certificate
-----END CERTIFICATE-----
```

接受的範例憑證（有效）：

1. 單一憑證 (PEM 編碼）：

```
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

2. 多個憑證 (PEM 編碼）：

```
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
# comments
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Base64-encoded certificate
-----END CERTIFICATE-----
```

HTTP 標頭和交互 TLS

本節說明使用交互 TLS 與用戶端交涉連線時，Application Load Balancer 用來傳送憑證資訊的 HTTP 標頭。Application Load Balancer 使用的特定`X-Amzn-Mtls`標頭取決於您指定的交互 TLS 模式：傳遞模式或驗證模式。

如需 Application Load Balancer 支援的其他 HTTP 標頭的相關資訊，請參閱 HTTP 標頭和 Application Load Balancer。

傳遞模式的 HTTP 標頭

對於傳遞模式中的交互 TLS，Application Load Balancer 使用以下標頭。

X-Amzn-Mtls-Clientcert

此標頭包含連線中呈現之整個用戶端憑證鏈的 URL 編碼 PEM 格式，並以 +=/做為安全字元。

標頭內容範例：

```
X-Amzn-Mtls-Clientcert: -----BEGIN%20CERTIFICATE-----%0AMIID<...reduced...>do0g
%3D%3D%0A-----END%20CERTIFICATE-----%0A-----BEGIN%20CERTIFICATE-----
%0AMIID1<...reduced...>3eZlyKA%3D%3D%0A-----END%20CERTIFICATE-----%0A
```

驗證模式的 HTTP 標頭

對於驗證模式中的交互 TLS，Application Load Balancer 使用以下標頭。

X-Amzn-Mtls-Clientcert-Serial-Number

此標頭包含分葉憑證序號的十六進位表示法。

標頭內容範例：

```
X-Amzn-Mtls-Clientcert-Serial-Number: 03A5B1
```

X-Amzn-Mtls-Clientcert-Issuer

此標頭包含發行者辨別名稱 (DN) 的 RFC2253 字串表示。

標頭內容範例：

```
X-Amzn-Mtls-Clientcert-Issuer:
 CN=rootcamtls.com,OU=rootCA,O=mTLS,L=Seattle,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Subject

此標頭包含主體辨別名稱 (DN) 的 RFC2253 字串表示法。

標頭內容範例：

```
X-Amzn-Mtls-Clientcert-Subject: CN=client_.com,OU=client-3,O=mTLS,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Validity

此標頭包含 `notBefore`和 `notAfter` 日期的 ISO8601 格式。

標頭內容範例：

```
X-Amzn-Mtls-Clientcert-Validity:
 NotBefore=2023-09-21T01:50:17Z;NotAfter=2024-09-20T01:50:17Z
```

X-Amzn-Mtls-Clientcert-Leaf

此標頭包含分葉憑證的 URL 編碼 PEM 格式，並以 +=/做為安全字元。

標頭內容範例：

```
X-Amzn-Mtls-Clientcert-Leaf: -----BEGIN%20CERTIFICATE-----%0AMIIG<...reduced...>NmrUlw
%0A-----END%20CERTIFICATE-----%0A
```

公告憑證授權單位 (CA) 主旨名稱

廣告憑證授權機構 (CA) 主體名稱透過協助用戶端判斷在交互 TLS 身分驗證期間接受哪些憑證，來增強身分驗證程序。

當您啟用公告 CA 主體名稱時，Application Load Balancer 會根據其關聯的信任存放區，公告其信任的憑證授權機構 (CAs) 主體名稱清單。當用戶端透過 Application Load Balancer 連線至目標時，用戶端會收到信任的 CA 主體名稱清單。

在 TLS 交握期間，當 Application Load Balancer 請求用戶端憑證時，它會在其憑證請求訊息中包含信任的 CA 辨別名稱 (DNs) 清單。這有助於用戶端選取符合公告 CA 主體名稱的有效憑證，簡化身分驗證程序並減少連線錯誤。

您可以在新的和現有的接聽程式上啟用廣告 CA 主體名稱。如需詳細資訊，請參閱新增 HTTPS 接聽程式。

Application Load Balancer 的連線日誌

Elastic Load Balancing 提供連線日誌，可擷取傳送至 Application Load Balancer 之請求的屬性。連線日誌包含的資訊包括用戶端 IP 地址和連接埠、用戶端憑證資訊、連線結果和正在使用的 TLS 密碼。然後，這些連線日誌可用於檢閱請求模式和其他趨勢。

若要進一步了解連線日誌，請參閱 Application Load Balancer 的連線日誌

在 Application Load Balancer 上設定交互 TLS

若要使用交互 TLS 傳遞模式，您只需將接聽程式設定為接受來自用戶端的任何憑證。當您使用交互 TLS 傳遞時，Application Load Balancer 會使用 HTTP 標頭將整個用戶端憑證鏈傳送至目標，這可讓您在應用程式中實作對應的身分驗證和授權邏輯。如需詳細資訊，請參閱建立 Application Load Balancer 的 HTTPS 接聽程式。

當您在驗證模式下使用交互 TLS 時，Application Load Balancer 會在負載平衡器交涉 TLS 連線時，為用戶端執行 X.509 用戶端憑證驗證。

若要使用交互 TLS 驗證模式，請執行下列動作：

- 建立新的信任存放區資源。
- 上傳您的憑證授權機構 (CA) 套件，以及選擇性的撤銷清單。
- 將信任存放區連接到設定為驗證用戶端憑證的接聽程式。

使用下列程序在 Application Load Balancer 上設定交互 TLS 驗證模式。

任務

- 建立信任存放區
- 關聯信任存放區
- 取代 CA 憑證套件
- 新增憑證撤銷清單
- 刪除憑證撤銷清單
- 刪除信任存放區

建立信任存放區

如果您在建立負載平衡器或接聽程式時新增信任存放區，信任存放區會自動與新的接聽程式建立關聯。否則，您必須自行將其與接聽程式建立關聯。

先決條件

- 若要建立信任存放區，您必須擁有憑證授權單位 (CA) 的憑證套件。

Console

下列範例使用 主控台的信任存放區部分建立信任存放區。或者，您可以在建立 HTTP 接聽程式時建立信任存放區。

建立信任存放區

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇信任存放區。
3. 選擇建立信任存放區。
4. 信任存放區組態

   a. 針對信任存放區名稱，輸入信任存放區的名稱。

   b. 針對憑證授權機構套件，輸入要使用的 ca 憑證套件的 Amazon S3 路徑。

   c. （選用）使用物件版本來選取舊版的 ca 憑證套件。否則，會使用目前的版本。
5. （選用）對於撤銷，您可以將憑證撤銷清單新增至信任存放區。

   a. 選擇新增 CRL，然後在 Amazon S3 中輸入憑證撤銷清單的位置。

   b. （選用）使用物件版本來選取憑證撤銷清單的先前版本。否則，會使用目前的版本。
6. （選用）展開信任存放區標籤，並為信任存放區輸入最多 50 個標籤。
7. 選擇建立信任存放區。

AWS CLI

建立信任存放區

使用 create-trust-store 命令。

```
aws elbv2 create-trust-store \
    --name my-trust-store \
    --ca-certificates-bundle-s3-bucket amzn-s3-demo-bucket \
    --ca-certificates-bundle-s3-key certificates/ca-bundle.pem
```

CloudFormation

建立信任存放區

定義 AWS::ElasticLoadBalancingV2::TrustStore 類型的資源。

```
Resources:
  myTrustStore:
    Type: 'AWS::ElasticLoadBalancingV2::TrustStore'
    Properties:
      Name: my-trust-store
      CaCertificatesBundleS3Bucket: amzn-s3-demo-bucket
      CaCertificatesBundleS3Key: certificates/ca-bundle.pem
```

關聯信任存放區

建立信任存放區之後，您必須將其與接聽程式建立關聯，Application Load Balancer 才能開始使用信任存放區。您只能有一個信任存放區與每個安全接聽程式相關聯，但一個信任存放區可以與多個接聽程式相關聯。

Console

您可以建立信任存放區與現有接聽程式的關聯，如下列程序所示。或者，您可以在建立 HTTPS 接聽程式時關聯信任存放區。如需詳細資訊，請參閱建立 HTTPS 接聽程式。

建立信任存放區的關聯

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選擇 Protocol：Port 欄中的連結，以開啟安全接聽程式的詳細資訊頁面。
5. 在安全索引標籤上，選擇編輯安全接聽程式設定。
6. 如果未啟用交互 TLS，請在用戶端憑證處理下選取相互身分驗證 (mTLS)，然後選擇使用信任存放區驗證。
7. 針對信任存放區，選擇信任存放區。
8. 選擇儲存變更。

AWS CLI

建立信任存放區的關聯

使用 modify-listener 命令。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --mutual-authentication "Mode=verify,TrustStoreArn=trust-store-arn"
```

CloudFormation

建立信任存放區的關聯

更新 AWS::ElasticLoadBalancingV2::Listener 資源。

```
Resources:
  myHTTPSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTPS
      Port: 443
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
      Certificates:
        - CertificateArn: certificate-arn
      MutualAuthentication:
        - Mode: verify
          TrustStoreArn: trust-store-arn
```

取代 CA 憑證套件

CA 憑證套件是信任存放區的必要元件。這是由憑證授權單位驗證的信任根憑證和中繼憑證集合。這些經過驗證的憑證可確保用戶端信任所呈現的憑證是由負載平衡器所擁有。

信任存放區一次只能包含一個 CA 憑證套件，但您可以在建立信任存放區之後隨時取代 CA 憑證套件。

Console

若要取代 CA 憑證套件

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇信任存放區。
3. 選取信任存放區。
4. 選擇動作、取代 CA 套件。
5. 在取代 CA 套件頁面的憑證授權單位套件下，輸入所需 CA 套件的 Amazon S3 位置。
6. （選用）使用物件版本來選取憑證撤銷清單的先前版本。否則，會使用目前的版本。
7. 選取取代 CA 套件。

AWS CLI

若要取代 CA 憑證套件

使用 modify-trust-store 命令。

```
aws elbv2 modify-trust-store \
    --trust-store-arn trust-store-arn \
    --ca-certificates-bundle-s3-bucket amzn-s3-demo-bucket-new \
    --ca-certificates-bundle-s3-key certificates/new-ca-bundle-pem
```

CloudFormation

更新 CA 憑證套件

定義 AWS::ElasticLoadBalancingV2::TrustStore 類型的資源。

```
Resources:
  myTrustStore:
    Type: 'AWS::ElasticLoadBalancingV2::TrustStore'
    Properties:
      Name: my-trust-store
      CaCertificatesBundleS3Bucket: amzn-s3-demo-bucket-new
      CaCertificatesBundleS3Key: certificates/new-ca-bundle.pem
```

新增憑證撤銷清單

或者，您可以為信任存放區建立憑證撤銷清單。撤銷清單由憑證授權單位發佈，並包含已撤銷憑證的資料。Application Load Balancer 僅支援 PEM 格式的憑證撤銷清單。

當憑證撤銷清單新增至信任存放區時，會為其提供撤銷 ID。新增至信任存放區的每個撤銷清單的撤銷 IDs都會增加，而且無法變更。

Application Load Balancer 無法撤銷憑證撤銷清單中序號為負的憑證。

Console

新增撤銷清單

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇信任存放區。
3. 選取信任存放區以檢視其詳細資訊頁面。
4. 在憑證撤銷清單索引標籤上，選取動作、新增撤銷清單。
5. 在新增撤銷清單頁面的憑證撤銷清單下，輸入所需憑證撤銷清單的 Amazon S3 位置

6. （選用）使用物件版本來選取憑證撤銷清單的先前版本。否則會使用目前的版本。
7. 選取新增撤銷清單

AWS CLI

新增撤銷清單

使用 add-trust-store-revocations 命令。

```
aws elbv2 add-trust-store-revocations \
    --trust-store-arn trust-store-arn \
    --revocation-contents "S3Bucket=amzn-s3-demo-bucket,S3Key=crl/revoked-
list.crl,RevocationType=CRL"
```

CloudFormation

新增撤銷清單

定義 AWS::ElasticLoadBalancingV2::TrustStoreRevocation 類型的資源。

```
Resources:
  myRevocationContents:
    Type: 'AWS:ElasticLoadBalancingV2::TrustStoreRevocation'
    Properties:
      TrustStoreArn: !Ref myTrustStore
      RevocationContents:
        - RevocationType: CRL
          S3Bucket: amzn-s3-demo-bucket
          S3Key: crl/revoked-list.crl
```

刪除憑證撤銷清單

當您不再需要憑證撤銷清單時，您可以將其刪除。當您從信任存放區刪除憑證撤銷清單時，也會刪除其撤銷 ID，而且在信任存放區的生命週期內不會重複使用。

Console

刪除撤銷清單

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格中，選擇信任存放區。
3. 選取信任存放區。
4. 在憑證撤銷清單索引標籤上，選擇動作、刪除撤銷清單。
5. 出現確認提示時，請按一下 **confirm**。
6. 選擇 刪除。

AWS CLI

刪除撤銷清單

使用 remove-trust-store-revocations 命令。

```
aws elbv2 remove-trust-store-revocations \
    --trust-store-arn trust-store-arn \
    --revocation-ids id-1 id-2 id-3
```

刪除信任存放區

當您不再為信任存放區使用 時，您可以將其刪除。您無法刪除與接聽程式相關聯的信任存放區。

Console

刪除信任存放區

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇信任存放區。
3. 選取信任存放區。
4. 選擇 刪除。
5. 出現確認提示時，請輸入 `confirm`，然後選擇 Delete (刪除)。

AWS CLI

刪除信任存放區

使用 delete-trust-store 命令。

```
aws elbv2 delete-trust-store \
```

```
    --trust-store-arn trust-store-arn
```

共用 Application Load Balancer 的 Elastic Load Balancing 信任存放區

Elastic Load Balancing 與 AWS Resource Access Manager (AWS RAM) 整合以啟用信任存放區共用。 AWS RAM 是一種服務，可讓您在組織或組織單位 (OUs) 之間 AWS 帳戶 和內部安全地共用 Elastic Load Balancing 信任存放區資源。如果您有多個帳戶，您可以建立信任存放區一次，並使用 AWS RAM 讓其他帳戶可以使用。如果您的帳戶由 管理 AWS Organizations，您可以與組織中的所有帳戶共用信任存放區，或僅與指定組織單位 (OUs內的帳戶共用信任存放區。

透過 AWS RAM，您可以透過建立資源共用來共用您擁有的資源。資源共享指定要共用的資源，以及共用它們的消費者。在此模型中，擁有信任存放區的 AWS 帳戶 （擁有者） 會與其他 AWS 帳戶 （取用者） 共用。消費者可以將共用信任存放區與其 Application Load Balancer 接聽程式建立關聯，方式與在自己的帳戶中建立信任存放區的關聯相同。

信任存放區擁有者可以與下列人員共用信任存放區：

- 中的特定組織 AWS 帳戶 內部或外部 AWS Organizations
- 中組織內部的組織單位 AWS Organizations
- 其在 中的整個組織 AWS Organizations

目錄

- 信任存放區共用的先決條件
- 共用信任存放區的許可
- 共用信任存放區
- 停止共用信任存放區
- 計費和計量

信任存放區共用的先決條件

- 您必須使用 建立資源共享 AWS Resource Access Manager。如需詳細資訊，請參閱AWS RAM 《 使用者指南》中的建立資源共享。
- 若要共用信任存放區，您必須在 中擁有信任存放區 AWS 帳戶。您無法共用已與您共用的信任存放區。

- 若要與組織或 中的組織單位共用信任存放區 AWS Organizations，您必須啟用與 共用 AWS Organizations。如需詳細資訊，請參閱《AWS RAM 使用者指南》中的透過 AWS Organizations啟用共用。

共用信任存放區的許可

信任存放區擁有者

- 信任存放區擁有者可以建立信任存放區。
- 信任存放區擁有者可以將信任存放區與相同帳戶中的負載平衡器搭配使用。
- 信任存放區擁有者可以與其他 AWS 帳戶或 共用信任存放區 AWS Organizations。
- 信任存放區擁有者可以從任何 AWS 帳戶或 取消共用信任存放區 AWS Organizations。
- 信任存放區擁有者無法防止負載平衡器在相同帳戶中使用信任存放區。
- 信任存放區擁有者可以使用共用信任存放區列出所有 Application Load Balancer。
- 如果沒有目前關聯，信任存放區擁有者可以刪除信任存放區。
- 信任存放區擁有者可以刪除與共用信任存放區的關聯。
- 使用共用信任存放區時，信任存放區擁有者會收到 CloudTrail 日誌。

信任存放區取用者

- 信任存放區取用者可以檢視共用信任存放區。
- 信任存放區消費者可以使用相同帳戶中的信任存放區來建立或修改接聽程式。
- 信任存放區取用者可以使用共用信任存放區建立或修改接聽程式。
- 信任存放區取用者無法使用不再共用的信任存放區建立接聽程式。
- 信任存放區取用者無法修改共用信任存放區。
- 當 與接聽程式相關聯時，信任存放區取用者可以檢視共用信任存放區 ARN。
- 信任存放區取用者在使用共用信任存放區建立或修改接聽程式時，會收到 CloudTrail 日誌。

受管許可

共用信任存放區時，資源共用會使用受管許可來控制信任存放區取用者允許的動作。您可以使用預設的受管許可 `AWSRAMPermissionElasticLoadBalancingTrustStore`，其中包含所有可用的許可，或建立您自己的客戶受管許可。`DescribeTrustStores`、

DescribeTrustStoreRevocations和 DescribeTrustStoreAssociations許可一律會啟用，且無法移除。

信任存放區資源共用支援下列許可：

elasticloadbalancing:CreateListener

可以將共用信任存放區連接到新的接聽程式。

elasticloadbalancing:ModifyListener

可以將共用信任存放區連接到現有的接聽程式。

elasticloadbalancing:GetTrustStoreCaCertificatesBundle

可以下載與共用信任存放區相關聯的 ca 憑證套件。

elasticloadbalancing:GetTrustStoreRevocationContent

可以下載與共用信任存放區相關聯的撤銷檔案。

elasticloadbalancing：DescribeTrustStores （預設）

可以列出所有擁有並與帳戶共用的信任存放區。

elasticloadbalancing：DescribeTrustStoreRevocations （預設）

可以列出指定信任存放區 arn 的所有撤銷內容。

elasticloadbalancing：DescribeTrustStoreAssociations （預設）

可以列出信任存放區取用者帳戶中與共用信任存放區相關聯的所有資源。

共用信任存放區

若要共用信任存放區，您必須將其新增至資源共用。資源共用是可讓您在 AWS 帳戶之間共用資源的一種 AWS RAM 資源。資源共用會指定要共用的資源、與其共用的消費者，以及委託人可執行的動作。當您使用 Amazon EC2 主控台共用信任存放區時，您可以將其新增至現有的資源共用。若要將信任存放區新增至新的資源共享，您必須先使用 AWS RAM 主控台建立資源共享。

當您與其他人共用您擁有的信任存放區時 AWS 帳戶，您可以讓這些帳戶將其 Application Load Balancer 接聽程式與帳戶中的信任存放區建立關聯。

如果您是 中組織的一部分， AWS Organizations 且已啟用組織內的共用，則組織中的消費者會自動獲得共用信任存放區的存取權。否則，消費者會收到加入資源共享的邀請，並在接受邀請後獲得共用信任存放區的存取權。

您可以使用 Amazon EC2 主控台、 AWS RAM 主控台或 共享您擁有的信任存放區 AWS CLI。

使用 Amazon EC2 主控台共享您擁有的信任存放區

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的負載平衡下，選擇信任存放區。
3. 選取信任存放區名稱以檢視其詳細資訊頁面。
4. 在共用索引標籤上，選擇共用信任存放區。
5. 在共用信任存放區頁面的資源共用下，選取要共用信任存放區的資源共用。
6. （選用）如果您需要建立新的資源共享，請選取在 RAM 主控台中建立資源共享連結。
7. 選取共用信任存放區。

使用 AWS RAM 主控台共享您擁有的信任存放區

請參閱《AWS RAM 使用者指南》中的建立資源共享。

使用 共享您擁有的信任存放區 AWS CLI

使用 create-resource-share 命令。

停止共用信任存放區

若要停止共用您擁有的信任存放區，您必須將其從資源共用中移除。在您停止共用信任存放區後，現有的關聯會保留，但不允許與先前共用信任存放區的新關聯。當信任存放區擁有者或信任存放區取用者刪除關聯時，會從兩個帳戶中刪除該關聯。如果信任存放區取用者想要離開資源共用，則必須要求資源共用的擁有者移除帳戶。

> **⚠ 刪除關聯**
>
> 信任存放區擁有者可以使用 DeleteTrustStoreAssociation 命令強制刪除現有的信任存放區關聯。刪除關聯時，任何使用信任存放區的負載平衡器接聽程式都無法再驗證用戶端憑證，而且 TLS 交握會失敗。

您可以使用 Amazon EC2 主控台、 AWS RAM 主控台或 停止共用信任存放區 AWS CLI。

使用 Amazon EC2 主控台停止共用您擁有的信任存放區

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在導覽窗格的負載平衡下，選擇信任存放區。
3. 選取信任存放區名稱以檢視其詳細資訊頁面。
4. 在共用索引標籤的資源共用下，選取要停止共用的資源共用。
5. 選擇移除。

使用 AWS RAM 主控台停止共用您擁有的信任存放區

請參閱《AWS RAM 使用者指南》中的更新資源共享。

使用 停止共用您擁有的信任存放區 AWS CLI

使用 disassociate-resource-share 命令。

計費和計量

共用信任存放區會產生相同的標準信任存放區費率，每小時計費，每個信任存放區與 Application Load Balancer 的關聯。

如需詳細資訊，包括每個區域的特定費率，請參閱 Elastic Load Balancing 定價

使用 Application Load Balancer 來驗證使用者身分

您可以設定 Application Load Balancer，以在使用者存取應用程式時安全地驗證使用者身分。這可讓您將驗證使用者的工作卸載到您的負載平衡器，使得您的應用程式可以專注在商業邏輯上。

支援下列使用案例：

- 透過與 OpenID Connect (OIDC) 相容的身分提供者 (IdP) 驗證使用者。
- 透過 Amazon Cognito 支援的使用者集區，經由社交 IdP (例如 Amazon、Facebook 或 Google) 驗證使用者身分。
- 透過 Amazon Cognito 支援的使用者集區，使用 SAML、OpenID Connect (OIDC) 或 OAuth，透過公司身分來驗證使用者身分。

準備使用 OIDC 合規 IdP

如果您使用 OIDC 合規 IdP 搭配 Application Load Balancer，請執行下列動作：

- 在 IdP 中建立新 OIDC 應用程式。IdP 的 DNS 必須可公開解析。
- 您必須設定用戶端 ID 和用戶端機密。

- 取得 IdP 發佈的以下端點：授權、字符和使用者資訊。您可以在此組態中找到此資訊。
- IdP 端點憑證的發行者應是受信任的公用憑證授權單位。
- 端點的 DNS 項目必須可公開解析，即使會解析為私有 IP 地址也可以。
- 允許以下其中一個重新導向 URL 加入您的使用者將使用的 IdP 應用程式，其中的 DNS 是您的負載平衡器的網域名稱，而 CNAME 為您的應用程式的 DNS 別名：
  - https://*DNS*/oauth2/idpresponse
  - https://*CNAME*/oauth2/idpresponse

準備使用 Amazon Cognito

可用的區域

Application Load Balancer 的 Amazon Cognito 整合可在下列區域使用：

- 美國東部 (維吉尼亞北部)
- 美國東部 (俄亥俄)
- 美國西部 (加利佛尼亞北部)
- 美國西部 (奧勒岡)
- 加拿大 (中部)
- 加拿大西部 (卡加利)
- 歐洲 (斯德哥爾摩)
- 歐洲 (米蘭)
- 歐洲 (法蘭克福)
- 歐洲 (蘇黎世)
- 歐洲 (愛爾蘭)
- 歐洲 (倫敦)
- Europe (Paris)
- 歐洲 (西班牙)
- 南美洲 (聖保羅)
- 亞太地區 (香港)
- 亞太地區 (東京)
- 亞太地區 (首爾)

- 亞太地區 (大阪)
- 亞太地區 (孟買)
- 亞太地區 (海德拉巴)
- 亞太地區 (新加坡)
- 亞太地區 (悉尼)
- 亞太地區 (雅加達)
- 亞太地區 (墨爾本)
- 中東 (阿拉伯聯合大公國)
- Middle East (Bahrain)
- 非洲 (開普敦)
- 以色列 (特拉維夫)

如果您使用與 Amazon Cognito 使用者集區搭配 Application Load Balancer，請執行下列動作：

- 建立使用者集區。如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的Amazon Cognito 使用者集區。
- 建立使用者集區用戶端。您必須設定用戶端來產生用戶端機密，使用代碼授予流程，並支援負載平衡器使用的相同 OAuth 範圍。如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的設定使用者集區應用程式用戶端。
- 建立使用者集區物件網域。如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的設定使用者集區網域。
- 驗證請求的範圍傳回 ID 字符。例如，預設範圍 `openid` 會傳回 ID 字符，但 `aws.cognito.signin.user.admin` 範圍則不會。
- 若要聯合社交或公司 IdP，請在聯合區段啟用 IdP。如需詳細資訊，請參閱《Amazon Cognito 開發人員指南》中的使用第三方身分提供者登入使用者集區。
- 允許 Amazon Cognito 的回呼 URL 欄位中使用以下重新導向 URL，其中 DNS 是負載平衡器的網域名稱，而 CNAME 為應用程式的 DNS 別名 (如果您有使用)：
  - https://*DNS*/oauth2/idpresponse
  - https://*CNAME*/oauth2/idpresponse
- 允許 IdP 應用程式的回呼 URL 中使用使用者集區網域。針對 IdP 使用此格式。例如：
  - https://*domain-prefix*.auth.*region*.amazoncognito.com/saml2/idpresponse
  - https：//*user-pool-domain*/saml2/idpresponse

應用程式用戶端設定中的回呼 URL 必須全都使用小寫字母。

若要讓使用者設定負載平衡器以使用 Amazon Cognito 來驗證使用者身分，您必須授予使用者呼叫 `cognito-idp:DescribeUserPoolClient` 動作的許可。

準備使用 Amazon CloudFront

如果您在 Application Load Balancer 前端使用 CloudFront 分佈，請啟用以下設定：

- 轉送請求標頭 (全部) – 確保 CloudFront 不快取所經驗證請求的回應。這可避免在驗證工作階段過期後從快取中提供回應。或者，若要在快取已啟用時降低此風險，CloudFront 分佈的擁有者可以將存活時間 (TTL) 值設為在驗證 Cookie 過期之前過期。
- 查詢字串轉送和快取 (全部) – 確保負載平衡器可存取向 IdP 驗證使用者身分時所需的查詢字串參數。
- Cookie 轉送 (全部) – 確保 CloudFront 將所有驗證 Cookie 轉送到負載平衡器。
- 搭配 Amazon CloudFront 設定 OpenID Connect (OIDC) 身分驗證時，請確保在整個連線路徑中一致地使用 HTTPS 連接埠 443。否則，身分驗證失敗可能會發生，因為用戶端 OIDC 重新導向 URLs 與原始產生的 URI 的連接埠號碼不相符。

設定使用者身分驗證

您透過為一個或多個接聽程式規則建立驗證動作來設定使用者身分驗證。`authenticate-cognito` 和 `authenticate-oidc` 動作類型僅支援使用 HTTPS 接聽程式。如需對應欄位的說明，請參閱 Elastic Load Balancing API Reference (版本 2015-12-01)中的 AuthenticateCognitoActionConfig 和 AuthenticateOidcActionConfig。

負載平衡器會傳送工作階段 Cookie 給用戶端，以維護驗證狀態。此 Cookie 永遠包含 `secure` 屬性，因為使用者驗證需要 HTTPS 接聽程式。此 Cookie 包含具有 CORS (跨來源資源共享) 請求的 `SameSite=None` 屬性。

對於支援需要獨立用戶端身分驗證之多個應用程式的負載平衡器，每個具有身分驗證動作的接聽程式規則都應具有唯一的 Cookie 名稱。這可確保用戶端在路由傳送至規則中指定的目標群組之前，一律會使用 IdP 進行驗證。

Application Load Balancer 不支援 URL 編碼的 Cookie 值。

依預設，`SessionTimeout` 欄位會設為 7 天。如果您需要較短的工作階段，您可以將工作階段逾時設定為最短至 1 秒。如需詳細資訊，請參閱工作階段逾時。

根據您的應用程式適當地設定 `OnUnauthenticatedRequest` 欄位。例如：

- 需要使用者使用社交或公司身分登入的應用程式 – 這是透過預設選項 (authenticate) 支援。如果使用者未登入，負載平衡器會將請求重新導向至 IdP 授權端點，並且 IdP 會提示使用者使用其使用者界面登入。
- 為登入的使用者提供個人化檢視或對未登入的使用者提供一般檢視的應用程式 – 若要支援這類應用程式，請使用 allow 選項。如果使用者已登入，負載平衡器會提供使用者宣告，而應用程式可提供個人化的檢視。如果使用者未登入，負載平衡器會轉送不帶使用者宣告的請求，而應用程式可提供一般檢視。
- 含有 JavaScript (每隔幾秒鐘載入一次) 的單一頁面應用程式 – 如果使用 deny 選項，負載平衡器會將「HTTP 401 未經授權」錯誤傳回給沒有身分驗證資訊的 AJAX 呼叫。但是，如果使用者具有已過期的身分驗證資訊，則負載平衡器會將用戶端重新導向至 IdP 授權端點。

負載平衡器必須能夠與 IdP 字符端點 (TokenEndpoint) 和 IdP 使用者資訊端點 (UserInfoEndpoint) 通訊。Application Load Balancer 僅在與這些端點通訊時支援 IPv4。如果您的 IdP 使用公有地址，請確保負載平衡器的安全群組和 VPC 的網路 ACLs 允許存取端點。使用內部負載平衡器或 IP 地址類型 時dualstack-without-public-ipv4，NAT 閘道可以讓負載平衡器與端點通訊。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 NAT 閘道基本概念。

使用以下 create-rule 命令來設定使用者驗證。

```
aws elbv2 create-rule \
    --listener-arn listener-arn \
    --priority 10 \
    --conditions Field=path-pattern,Values="/login" \
    --actions file://actions.json
```

以下是指定 authenticate-oidc 動作和 forward 動作的 actions.json 檔案範例。AuthenticationRequestExtraParams 可讓您在身分驗證期間將額外的參數傳遞至 IdP。請依照身分提供者提供的文件來判斷受支援的欄位

```
[{
    "Type": "authenticate-oidc",
    "AuthenticateOidcConfig": {
        "Issuer": "https://idp-issuer.com",
        "AuthorizationEndpoint": "https://authorization-endpoint.com",
        "TokenEndpoint": "https://token-endpoint.com",
        "UserInfoEndpoint": "https://user-info-endpoint.com",
        "ClientId": "abcdefghijklmnopqrstuvwxyz123456789",
        "ClientSecret": "12345678901234567890123456789O",
        "SessionCookieName": "my-cookie",
```

```
        "SessionTimeout": 3600,
        "Scope": "email",
        "AuthenticationRequestExtraParams": {
            "display": "page",
            "prompt": "login"
        },
        "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
},
{
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
    "Order": 2
}]
```

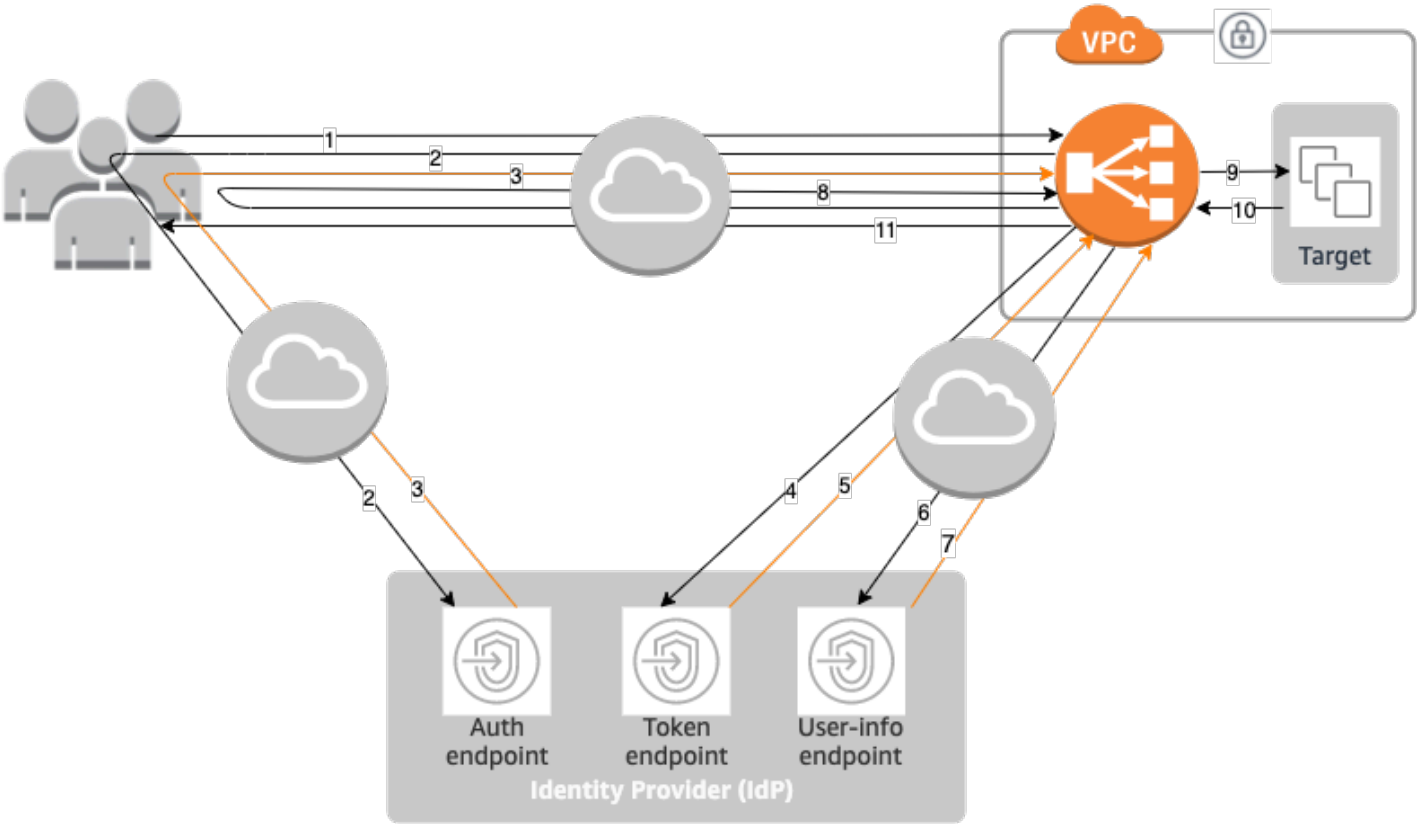
下列是會指定 authenticate-cognito 動作和 forward 動作的 actions.json 檔案的範例。

```
[{
    "Type": "authenticate-cognito",
    "AuthenticateCognitoConfig": {
        "UserPoolArn": "arn:aws:cognito-idp:region-code:account-id:userpool/user-pool-
id",
        "UserPoolClientId": "abcdefghijklmnopqrstuvwxyz123456789",
        "UserPoolDomain": "userPoolDomain1",
        "SessionCookieName": "my-cookie",
        "SessionTimeout": 3600,
        "Scope": "email",
        "AuthenticationRequestExtraParams": {
            "display": "page",
            "prompt": "login"
        },
        "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
},
{
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
    "Order": 2
}]
```

如需詳細資訊，請參閱Application Load Balancer 的接聽程式規則。

身分驗證流程

下列網路圖是 Application Load Balancer 如何使用 OIDC 驗證使用者身分的視覺化表示。



下面的帶編號項目著重解釋在前述網路圖中顯示的元素。

1. 使用者將 HTTPS 請求傳送至在 Application Load Balancer 後方託管的網站。當規則的條件滿足某個驗證動作時，負載平衡器會請求標頭中檢查身分驗證工作階段 Cookie。
2. 如果 Cookie 不存在，負載平衡器會將使用者重新導向到 IdP 授權端點，使得 IdP 可驗證使用者。
3. 在使用者通過身分驗證之後，IdP 會將使用者重新傳送回負載平衡器並帶有一個授權授予代碼。
4. 負載平衡器會向 IdP 權杖端點提供授權授予代碼。
5. 在收到有效的授權授予代碼後，IdP 會向 Application Load Balancer 提供 ID 權杖和存取權杖。
6. 然後，Application Load Balancer 會將存取權杖傳送到使用者資訊端點。
7. 使用者資訊端點會將存取權杖交換為使用者宣告。
8. Application Load Balancer 會將具有 AWSELB 身分驗證工作階段 Cookie 的使用者重新導向至原始 URI。由於大部分瀏覽器會將 Cookie 大小限制在 4K，負載平衡器會將大小大於 4K 的 Cookie 分成多個 Cookie 碎片。如果從 IdP 收到的使用者宣告和存取字符的總大小大於 11K 位元組，負

載平衡器會傳回 HTTP 500 錯誤給用戶端，並且遞增 ELBAuthUserClaimsSizeExceeded 指標。

9. Application Load Balancer 會驗證 Cookie，並將使用者資訊轉送至 X-AMZN-OIDC-* HTTP 標頭集中的目標。如需詳細資訊，請參閱使用者宣告編碼和簽章驗證。
10. 此目標會將回應傳送至 Application Load Balancer。
11. Application Load Balancer 會將最終回應傳送給使用者。

每個新的請求都會經歷步驟 1 到 11，而後續的請求則會經歷步驟 9 到 11。也就是說，只要 Cookie 尚未過期，每個後續請求都會從步驟 9 開始。

使用者在 IdP 進行身分驗證之後，系統會將 AWSALBAuthNonce Cookie 新增至請求標頭。這並不會變更 Application Load Balancer 處理來自 IdP 重新導向請求的方式。

如果 IdP 在 ID 字符中提供有效的重新整理字符，負載平衡器會儲存該重新整理字符，並在每次存取字符過期時使用它來重新整理使用者宣告，直到工作階段逾時或 IdP 重新整理失敗。如果使用者登出，重新整理會失敗，並且負載平衡器會將使用者重新導向至 IdP 授權端點。這可讓負載平衡器在使用者登出後捨棄工作階段。如需詳細資訊，請參閱工作階段逾時。

> **Note**
>
> Cookie 到期時間與身分驗證工作階段到期時間不同。Cookie 到期時間是 Cookie 的一種屬性，設為 7 天。身分驗證工作階段的實際長度取決於在 Application Load Balancer 上為身分驗證功能設定的工作階段逾時。此工作階段逾時包含在 Auth Cookie 值中，該值也經過加密。

使用者宣告編碼和簽章驗證

負載平衡器成功驗證使用者之後，它會將從 IdP 收到的使用者宣告傳送到目標。負載平衡器會簽署使用者宣告，讓應用程式可以驗證簽章並驗證宣告是由負載平衡器傳送。

負載平衡器會新增下列 HTTP 標頭：

`x-amzn-oidc-accesstoken`

來自字符端點的存取字符，純文字格式。

`x-amzn-oidc-identity`

來自使用者資訊端點的主旨欄位 (sub)，純文字格式。

注意：子宣告是識別特定使用者的最佳方法。

`x-amzn-oidc-data`

使用者宣告，JSON Web 字符 (JWT) 格式。

存取權杖和使用者宣告與 ID 權杖不同。存取權杖和使用者宣告僅允許存取伺服器資源，而 ID 權杖會附帶其他資訊來對使用者進行身分驗證。Application Load Balancer 在驗證使用者時建立新的存取權杖，而且只會將存取權杖和宣告傳遞給後端，但不會傳遞 ID 權杖資訊。

這些字符採用 JWT 格式，但不是 ID 字符。JWT 格式包含使用 base64 URL 編碼的標頭、承載和簽章，而且尾端包含填補字元。Application Load Balancer 會使用 ES256 (使用 P-256 和 SHA256 的 ECDSA) 來產生 JWT 簽章。

JWT 標頭是 JSON 物件，具有下列欄位：

```
{
   "alg": "algorithm",
   "kid": "12345678-1234-1234-1234-123456789012",
   "signer": "arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/
app/load-balancer-name/load-balancer-id",
   "iss": "url",
   "client": "client-id",
   "exp": "expiration"
}
```

JWT 承載為 JSON 物件，其中包含從 IdP 使用者資訊端點收到的使用者宣告。

```
{
   "sub": "1234567890",
   "name": "name",
   "email": "alias@example.com",
   ...
}
```

如果您希望負載平衡器加密您的使用者宣告，您必須將目標群組設定為使用 HTTPS。此外，根據安全最佳實務，建議您將目標限制為僅接收來自 Application Load Balancer 的流量。您可以透過設定目標的安全群組來參考負載平衡器的安全群組 ID 來達成此目的。

為了確保安全性，您必須先驗證簽章，才能根據宣告進行任何授權，並驗證 JWT 標頭中的 `signer` 欄位是否包含預期的 Application Load Balancer ARN。

若要取得公有金鑰，請從 JWT 標頭取得金鑰 ID，並用其在端點查閱公有金鑰。每個 AWS 區域的端點如下：

```
https://public-keys.auth.elb.region.amazonaws.com/key-id
```

對於 AWS GovCloud (US)，端點如下所示：

```
https://s3-us-gov-west-1.amazonaws.com/aws-elb-public-keys-prod-us-gov-west-1/key-id
https://s3-us-gov-east-1.amazonaws.com/aws-elb-public-keys-prod-us-gov-east-1/key-id
```

AWS 提供程式庫，可用來驗證由 Amazon Cognito、Application Load Balancer 和其他 OIDC 相容 IDPs JWTs。如需詳細資訊，請參閱 AWS JWT Verify。

Timeout (逾時)

工作階段逾時

重新整理字符和工作階段逾時的共同運作方式如下所示：

- 如果工作階段逾時短於存取字符過期，負載平衡器會採用該工作階段逾時。如果使用者有 IdP 的主動工作階段，應該就不會提示使用者再次登入。否則，系統會將使用者重新導向至登入。
  - 如果 IdP 工作階段逾時較 Application Load Balancer 工作階段逾時更久，則使用者不需要提供憑證即可重新登入。而且，IdP 會使用新的授權授予代碼重新導向至 Application Load Balancer。即使沒有重新登入，授權碼也只能使用一次。
  - 如果 IdP 工作階段逾時較 Application Load Balancer 工作階段逾時一樣久或更久，則會要求使用者提供憑證以重新登入。使用者登入後，IdP 會使用新的授權授予代碼重新導向至 Application Load Balancer，其餘的身分驗證流程會繼續進行，直到請求到達後端為止。
- 如果工作階段逾時較存取權杖過期時間更久，並且 IdP 不支援重新整理權杖，負載平衡器會保留該身分驗證工作階段，直到其逾時為止。然後，負載平衡器會要求使用者再次登入。
- 如果工作階段逾時為長於存取字符過期，並且 IdP 支援重新整理字符，負載平衡器會在每次存取字符過期時重新整理該使用者工作階段。負載平衡器只會在驗證工作階段逾或重新整理流程失敗時，才會將使用者再次登入。

用戶端登入逾時

用戶端必須在 15 分鐘內啟動並完成身分驗證程序。如果用戶端無法在 15 分鐘的限制內完成身分驗證，則會從負載平衡器收到 HTTP 401 錯誤。此逾時限制無法變更或移除。

例如，如果使用者透過 Application Load Balancer 載入登入頁面，則必須在 15 分鐘內完成登入程序。如果使用者在 15 分鐘逾時過期後嘗試登入，負載平衡器會傳回 HTTP 401 錯誤。使用者必須重新整理頁面並嘗試再次登入。

身分驗證登出

當應用程式需要將已驗證的使用者登出時，您應該將驗證工作階段 Cookie 的過期時間設定為 -1，並將用戶端重新導向至 IdP 登出端點 (如果 IdP 有支援)。為了防止使用者重複使用已刪除的 Cookie，建議您將存取字符設定為合理的簡短過期時間。如果用戶端為負載平衡器提供具有過期存取字符和非 NULL 重新整理字符的工作階段 Cookie，負載平衡器會聯絡 IdP 以確定使用者是否仍然登入。

用戶端登出登陸頁面未經驗證。這表示它們不能位於需要身分驗證的 Application Load Balancer 規則後方。

- 請求傳送至目標後，應用程式必須將所有身分驗證 Cookie 的到期時間設定為 -1。Application Load Balancer 支援的 Cookie 大小上限為 16K，因此最多會建立 4 個碎片以傳送給用戶端。
  - 如果 IdP 具有登出端點，則應發出一個至 IdP 登出端點 (例如《Amazon Cognito 開發人員指南》中記錄的登出端點) 的重新導向。
  - 如果 IdP 沒有登出端點，請求會回到用戶端登出登陸頁面，並登入程序會重新啟動。
- 假設 IdP 具有登出端點，則 IdP 必須使存取權杖和重新整理權杖過期，並將使用者重新導向回用戶端登出登陸頁面。
- 後續請求會遵循原始身分驗證流程。

使用 Application Load Balancer 驗證 JWTs

您可以設定 Application Load Balancer (ALB) 來驗證用戶端提供的 JSON Web Token (JWT)，以進行安全service-to-service(S2S) machine-to-machine(M2M) 通訊。無論 JWT 的發出方式為何，負載平衡器都可以驗證 JWT，無需人為互動。

ALB 將驗證字符簽章，並需要兩個強制性宣告：'iss'（發行者）和 'exp'（過期）。此外，如果字符中存在，ALB 也會驗證 'nbf'（而不是之前）和 'iat'（在時間發出）宣告。您最多可以設定 10 個額外的宣告進行驗證。這些宣告支援三種格式：

- 單一字串：單一文字值
- 空格分隔值：以空格分隔的多個值（最多 10 個值）
- String-array：文字值的陣列（最多 10 個值）

如果權杖有效，負載平衡器會像往目標一樣使用權杖轉送請求。否則，它會拒絕請求。

準備使用 JWT 驗證

完成下列任務：

1. 向 IdP 註冊您的服務，IdP 會發出用戶端 ID 和用戶端秘密。
2. 單獨呼叫 IdP 以請求存取服務。IdP 會以存取字符回應。此字符通常是由 IdP 簽署的 JWT。
3. 設定 JSON Web 金鑰集 (JWKS) 端點。負載平衡器會在您設定的已知位置取得 IdP 發佈的公有金鑰。
4. 在請求標頭中包含 JWT，並在每個請求中將其轉送至 Application Load Balancer。注意：僅支援 RS256 演算法

JWT 驗證限制

搭配 Application Load Balancer 使用 JWT 驗證時，JWKS (JSON Web 金鑰集） 端點必須符合下列要求：

- 最大回應大小：150 KB
- 金鑰數量上限：10 個金鑰

如果身分提供者的 JWKS 回應超過其中一個限制，Application Load Balancer 不會將請求轉送到您的後端目標。

如果身分提供者的 JWKS 端點超過這些限制，請考慮在應用程式程式碼中實作 JWT 驗證，或使用具有較小金鑰集的身分提供者。

使用主控台設定 JWT 驗證

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡器)，選擇 Load Balancer (負載平衡器)。
3. 選取您的 Application Load Balancer，然後選擇接聽程式索引標籤。
4. 選取 HTTPS 接聽程式，然後選擇管理規則。
5. 選擇新增規則。
6. （選用） 若要指定規則的名稱，請展開名稱和標籤，然後輸入名稱。若要新增其他標籤，請選擇新增其他標籤，然後輸入標籤索引鍵和標籤值。

7. 在條件下，定義 1-5 個條件值
8. （選用）若要新增轉換，請選擇新增轉換、選擇轉換類型，然後輸入要比對的規則運算式和替代字串。
9. 針對動作、預先路由動作，選擇驗證字符。
   a. 對於 JWKS 端點，輸入 JSON Web 金鑰集端點的 URL。此端點必須可公開存取，並傳回用於驗證 JWT 簽章的公有金鑰。
   b. 對於發行者，在您的 JWT 權杖中輸入 iss 宣告的預期值。
   c. （選用）若要驗證其他宣告，請選擇其他宣告。
      i. 針對宣告名稱，輸入要驗證的宣告名稱。
      ii. 針對格式，選擇應如何解譯宣告值：
         1. 單一字串：宣告必須完全符合一個指定的值。
         2. 字串陣列：宣告必須符合陣列中的其中一個值。
         3. 空格分隔值：宣告包含空格分隔值，必須包含指定的值。
      iii. 在值中，輸入宣告的預期值。
      iv. 針對其他宣告重複（最多 10 個宣告）。
10. 對於動作、路由動作，選取成功驗證權杖後應執行的主要動作（轉送、重新導向或傳回固定回應）。
11. 視需要設定主要動作
12. 選擇儲存。

使用 CLI 設定 JWT 驗證

使用下列 create-rule 命令來設定 JWT 驗證 。

使用 動作建立接聽程式規則來驗證 JWTs。接聽程式必須是 HTTPS 接聽程式。

> **Note**
>
> 設定 JWT 驗證時，請確保 JWKS 端點回應的大小不超過 150 KB 或包含超過 10 個金鑰。超過這些限制的回應將阻止請求轉送到您的目標。

```
aws elbv2 create-rule \
```

```
    --listener-arn listener-arn \
    --priority 10 \
    --conditions Field=path-pattern,Values="/login" \
    --actions file://actions.json
```

以下是指定 jwt-validation動作和 forward動作的 actions.json 檔案範例。請依照身分提供者提供的文件來判斷受支援的欄位

```
--actions '[
    {
        "Type":"jwt-validation",
        "JwtValidationConfig":{
            "JwksEndpoint":"https://issuer.example.com/.well-known/jwks.json",
            "Issuer":"https://issuer.com"
        },
        "Order":1
    },
    {
        "Type":"forward",
        "TargetGroupArn":"target-group-arn",
        "Order":2
    }
]'
```

下列範例會指定要驗證的其他宣告。

```
--actions '[
    {
        "Type":"jwt-validation",
        "JwtValidationConfig":{
            "JwksEndpoint":"https://issuer.example.com/.well-known/jwks.json",
            "Issuer":"https://issuer.com",
            "AdditionalClaims":[
              {
                  "Format":"string-array",
                  "Name":"claim_name",
                  "Values":["value1","value2"]
              }
            ],
        },
        "Order":1
    },
    {
```

```
        "Type":"forward",
        "TargetGroupArn":"target-group-arn",
        "Order":2
    }
]'
```

如需詳細資訊，請參閱the section called “接聽程式規則”。

HTTP 標頭和 Application Load Balancer

HTTP 請求和 HTTP 回應使用標頭欄位來傳送有關 HTTP 訊息的資訊。HTTP 標頭會自動新增。標頭欄位是以冒號分隔的名稱值組，以歸位字元 (CR) 和換行 (LF) 分隔。一組以 RFC 2616 定義的標準 HTTP 標頭欄位，訊息標頭。也有應用程式廣泛採用的非標準 HTTP 標頭可用 (而且會自動新增)。有些非標準 HTTP 標頭擁有 `X-Forwarded` 字首。Application Load Balancer 支援以下 `X-Forwarded` 標頭。

如需 HTTP 連線的詳細資訊，請參閱 Elastic Load Balancing 使用者指南中的請求路由。

X-Forwarded 標頭

- X-Forwarded-For
- X-Forwarded-Proto
- X-Forwarded-Port

X-Forwarded-For

當您使用 HTTP 或 HTTPS 負載平衡器時，`X-Forwarded-For` 請求標頭會協助您識別用戶端的 IP 地址。由於負載平衡器攔截用戶端和伺服器之間的流量，伺服器存取日誌僅包含負載平衡器的 IP 地址。若要查看用戶端的 IP 地址，請使用 `routing.http.xff_header_processing.mode` 屬性。此屬性可讓您在 Application Load Balancer 將 HTTP 請求傳送至目標之前，修改、保留或移除該請求中的 `X-Forwarded-For` 標頭。此屬性的可能值為 `append`、`preserve` 和 `remove`。此屬性的預設值為 `append`。

> **Important**
>
> 由於可能存在安全風險，因此應謹慎使用 `X-Forwarded-For`標頭。只有在由網路內適當保護的系統新增時，才能將項目視為值得信任。

處理模式

- 附加
- 保留
- 移除

附加

Application Load Balancer 依預設會將用戶端的 IP 地址儲存在 X-Forwarded-For 請求標頭，並將標頭傳遞給伺服器。如果 X-Forwarded-For 請求標頭未包含在原始請求中，負載平衡器會以用戶端 IP 地址作為請求值建立請求標頭。否則，負載平衡器會將用戶端 IP 地址附加到現有的標頭，然後將標頭傳遞到您的伺服器。X-Forwarded-For 請求標頭可能包含以逗號分隔的多個 IP 地址。

X-Forwarded-For 請求標頭採用以下格式：

```
X-Forwarded-For: client-ip-address
```

下列是具有 IP 地址 203.0.113.7 之用戶端的範例 X-Forwarded-For 請求標頭。

```
X-Forwarded-For: 203.0.113.7
```

下列是具有 IPv6 地址 2001:DB8::21f:5bff:febf:ce22:8a2e 之用戶端的範例 X-Forwarded-For 請求標頭。

```
X-Forwarded-For: 2001:DB8::21f:5bff:febf:ce22:8a2e
```

負載平衡器上啟用用戶端連接埠保留屬性 (routing.http.xff_client_port.enabled) 後，X-Forwarded-For 請求標頭會包含在 client-ip-address 附加的 client-port-number (以冒號分隔)。標頭採用以下格式：

```
IPv4 -- X-Forwarded-For: client-ip-address:client-port-number
```

```
IPv6 -- X-Forwarded-For: [client-ip-address]:client-port-number
```

請注意，對於 IPv6，當負載平衡器將 client-ip-address 附加到現有的標頭時，其會以方括號括住該地址。

下列是用戶端 (IPv4 地址為 12.34.56.78，連接埠號碼為 8080) 的 X-Forwarded-For 請求標頭範例。

```
X-Forwarded-For: 12.34.56.78:8080
```

下列是用戶端 (IPv6 地址為 2001:db8:85a3:8d3:1319:8a2e:370:7348，連接埠號碼為 8080) 的 X-Forwarded-For 請求標頭範例。

```
X-Forwarded-For: [2001:db8:85a3:8d3:1319:8a2e:370:7348]:8080
```

保留

屬性中的 preserve 模式可確保 HTTP 請求的 X-Forwarded-For 標頭在傳送到目標之前，請求中不會以任何方式遭到修改。

移除

屬性中的 remove 模式會在將 HTTP 請求的 X-Forwarded-For 標頭在傳送到目標之前將其移除。

如果您啟用用戶端連接埠保留屬性 (routing.http.xff_client_port.enabled)，並為 routing.http.xff_header_processing.mode 屬性選取 preserve 或 remove，則 Application Load Balancer 會覆寫用戶端連接埠保留屬性。此屬性可將 X-Forwarded-For 標頭保持不變，或者根據您選取的模式將其移除，然後再將標頭傳送到目標。

下表顯示當您選取 append、preserve 或 remove 模式時，目標接收到的 X-Forwarded-For 標頭範例。在此範例中，最後一個跳轉的 IP 地址為 127.0.0.1。

請求說明	範例請求	append	preserve	remove
傳送的請求沒有 XFF 標頭	GET / index.html HTTP/1.1 Host: example.com	X-Forwarded-For: 127.0.0.1	不存在	不存在
傳送的請求包含 XFF 標頭和用戶端 IP 地址。	GET / index.html HTTP/1.1 Host: example.com X-Forward	X-Forwarded-For: 127.0.0.4, 127.0.0.1	X-Forwarded-For: 127.0.0.4	不存在

請求說明	範例請求	append	preserve	remove
	`ed-For: 127.0.0.4`			
傳送的請求包含 XFF 標頭和多個用戶端 IP 地址。	`GET / index.html HTTP/1.1 Host: example.com X-Forwarded-For: 127.0.0.4, 127.0.0.8`	`X-Forwarded-For: 127.0.0.4, 127.0.0.8, 127.0.0.1`	`X-Forwarded-For: 127.0.0.4, 127.0.0.8`	不存在

Console

管理 X-Forwarded-For標頭

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在屬性索引標籤中，選擇編輯。
5. 在流量組態區段的封包處理下，針對 X-Forwarded-For 標頭，選擇附加 （預設）、保留或移除。
6. 選擇儲存變更。

AWS CLI

管理 X-Forwarded-For標頭

以 屬性來使用 modify-load-balancer-attributes`routing.http.xff_header_processing.mode` 命令。可能的值為 `append`、`preserve` 和 `remove`。預設值為 append。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
```

```
    --attributes "Key=routing.http.xff_header_processing.mode,Value=preserve"
```

CloudFormation

管理 X-Forwarded-For標頭

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含 routing.http.xff_header_processing.mode 屬性。可能的值為 append、preserve 和 remove。預設值為 append。

```
Resources:
  myLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "routing.http.xff_header_processing.mode"
          Value: "preserve"
```

X-Forwarded-Proto

X-Forwarded-Proto 請求標頭協助您識別用戶端用於連接到您的負載平衡器的通訊協定 (HTTP 或 HTTPS)。您的伺服器存取日誌僅包含在伺服器和負載平衡器之間使用的通訊協定，但不包含用戶端和負載平衡器之間使用的通訊協定相關資訊。若要判斷用戶端和負載平衡器之間使用的通訊協定，請使用 X-Forwarded-Proto 請求標頭。Elastic Load Balancing 會將用戶端和負載平衡器之間使用的通訊協定儲存在 X-Forwarded-Proto 請求標頭，並將標頭傳遞給您的伺服器。

您的應用程式或網站可以使用存放在 X-Forwarded-Proto 請求標頭中的通訊協定，藉以產生重新導向到適當的 URL 的回應。

X-Forwarded-Proto 請求標頭採用以下格式：

```
X-Forwarded-Proto: originatingProtocol
```

以下範例包含適用於從用戶端產生的 X-Forwarded-Proto 請求標頭，以做為 HTTPS 請求：

```
X-Forwarded-Proto: https
```

X-Forwarded-Port

X-Forwarded-Port 請求標頭協助您識別用戶端用於連接到負載平衡器的目的地連接埠。

Application Load Balancer 的 HTTP 標頭修改

Application Load Balancer 支援對請求和回應標頭進行 HTTP 標頭修改。無需更新應用程式程式碼，標頭修改可讓您進一步控制應用程式的流量和安全性。

若要啟用標頭修改，請參閱 啟用標頭修改。

重新命名 mTLS/TLS 標頭

標頭重新命名功能可讓您設定 Application Load Balancer 產生並新增至請求的 mTLS 和 TLS 標頭名稱。

修改 HTTP 標頭的功能可讓您的 Application Load Balancer 輕鬆支援使用特定格式化請求和回應標頭的應用程式。

標頭	Description
X-Amzn-Mtls-Clientcert-Serial-Number	確保目標可以識別和驗證用戶端在 TLS 交握期間提供的特定憑證。
X-Amzn-Mtls-Clientcert-Issuer	透過識別發出憑證的憑證授權單位，協助目標驗證和驗證用戶端憑證。
X-Amzn-Mtls-Clientcert-Subject	提供目標有關用戶端憑證所發行實體的詳細資訊，這有助於在 mTLS 身分驗證期間進行識別、身分驗證、授權和記錄。
X-Amzn-Mtls-Clientcert-Validity	允許目標驗證正在使用的用戶端憑證是否在其定義的有效期間內，確保憑證不會過期或過早使用。

標頭	Description
X-Amzn-Mtls-Clientcert-Leaf	提供 mTLS 交握中使用的用戶端憑證，允許伺服器驗證用戶端並驗證憑證鏈。這可確保連線安全且獲得授權。
X-Amzn-Mtls-Clientcert	攜帶完整的用戶端憑證。允許目標驗證憑證的真實性、驗證憑證鏈，以及在 mTLS 交握程序期間驗證用戶端。
X-Amzn-TLS-Version	指出用於連線的 TLS 通訊協定版本。它有助於判斷通訊的安全層級、疑難排解連線問題並確保合規性。
X-Amzn-TLS-Cipher-Suite	指出用於保護 TLS 中連線的密碼編譯演算法組合。這可讓伺服器評估連線的安全性、協助進行相容性疑難排解，並確保符合安全政策。

新增回應標頭

您可以使用插入標頭，設定 Application Load Balancer 將安全相關的標頭新增至回應。透過這些屬性，您可以插入標頭，包括 HSTS、CHS 和 CSP。

根據預設，這些標頭為空白。發生這種情況時，Application Load Balancer 不會修改此回應標頭。

當您啟用回應標頭時，Application Load Balancer 會將具有設定值的標頭新增至所有回應。如果目標的回應包含 HTTP 回應標頭，負載平衡器會將標頭值更新為設定的值。否則，負載平衡器會將 HTTP 回應標頭新增至具有設定值的回應。

標頭	Description
Strict-Transport-Security	在指定的持續時間內，由瀏覽器強制執行僅限 HTTPS 的連線，協助防止man-in-the-middle攻擊、通訊協定降級和使用者錯誤。 可確保用戶端和目標之間的所有通訊都已加密。

標頭	Description
Access-Control-Allow-Origin	控制是否可以從不同的原始伺服器存取目標上的資源。這允許安全的跨來源互動，同時防止未經授權的存取。
Access-Control-Allow-Methods	指定向目標提出跨來源請求時允許的 HTTP 方法。它提供控制哪些動作可以從不同的原始伺服器執行。
Access-Control-Allow-Headers	指定哪些自訂或非簡單標頭可以包含在跨來源請求中。此標頭可讓目標控制來自不同原始伺服器的用戶端可以傳送哪些標頭。
Access-Control-Allow-Credentials	指定用戶端是否應在跨來源請求中包含憑證，例如 Cookie、HTTP 身分驗證或用戶端憑證。
Access-Control-Expose-Headers	允許目標指定用戶端可以在跨來源請求中存取哪些額外的回應標頭。
Access-Control-Max-Age	定義瀏覽器可以快取預檢請求結果的時間長度，減少重複預檢檢查的需求。這有助於透過減少特定跨來源請求所需的 OPTIONS 請求數量來最佳化效能。
Content-Security-Policy	透過控制指令碼、樣式、映像等資源，防止 XSS 等程式碼注入攻擊的安全功能可由網站載入和執行。
X-Content-Type-Options	使用無嗅探指令， 可透過防止瀏覽器猜測資源的 MIME 類型來增強 Web 安全性。它可確保瀏覽器僅根據宣告的內容類型解譯內容
X-Frame-Options	標頭安全機制，可透過控制網頁是否可以內嵌在框架中，協助防止點擊劫持攻擊。DENY 和 SAMEORIGIN 等值可確保內容不會內嵌在惡意或不受信任的網站。

停用標頭

使用停用標頭，您可以設定 Application Load Balancer 從回應中停用`server:awselb/2.0`標頭。這可減少伺服器特定資訊的暴露，同時為您的應用程式新增額外的保護層。

屬性名稱為 `routing.http.response.server.enabled`。可用的值為 `true`或 `false`。預設值為 `true`。

限制

- 標頭值可包含下列字元
  - 英數字元：a-z、 A-Z和 0-9
  - 特殊字元： _ :;.,\/'?!(){}[]@<>=-+*#&`|~^%
- 屬性的值大小不可超過 1K 位元組。
- Elastic Load Balancing 會執行基本輸入驗證，以確認標頭值是否有效。不過，驗證無法確認特定標頭是否支援該值。
- 為任何屬性設定空值會導致 Application Load Balancer 還原為預設行為。

為您的 Application Load Balancer 啟用 HTTP 標頭修改

標頭修改預設為關閉，且必須在每個接聽程式上啟用。如需詳細資訊，請參閱HTTP 標頭修改。

Console

啟用標頭修改

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取 Application Load Balancer。
4. 在接聽程式和規則索引標籤上，選取通訊協定和連接埠以開啟接聽程式的詳細資訊頁面。
5. 在屬性索引標籤上，選取編輯。

   接聽程式屬性會組織成群組。您將選擇要啟用哪些功能。
6. 【HTTPS 接聽程式】 可修改的 mTLS/TLS 標頭名稱

   a. 展開可修改的 mTLS/TLS 標頭名稱。

b. 啟用請求標頭來修改並提供其名稱。如需詳細資訊，請參閱the section called “重新命名 mTLS/TLS 標頭”。

7. 新增回應標頭

a. 展開 新增回應標頭。

b. 啟用回應標頭來新增並提供值。如需詳細資訊，請參閱the section called “新增回應標頭”。

8. ALB 伺服器回應標頭

- 啟用或停用伺服器標頭。

9. 選擇儲存變更。

AWS CLI

啟用標頭修改

使用 modify-listener-attributes 命令。如需屬性清單，請參閱 the section called “標頭修改屬性”。

```
aws elbv2 modify-listener-attributes \
    --listener-arn listener-arn \
    --attributes "Key=attribute-name,Value=attribute-value"
```

CloudFormation

啟用標頭修改

更新 AWS::ElasticLoadBalancingV2::Listener 資源以包含屬性。如需屬性清單，請參閱 the section called “標頭修改屬性”。

```
Resources:
  myHTTPlistener:
  Type: 'AWS::ElasticLoadBalancingV2::Listener'
  Properties:
    LoadBalancerArn: !Ref myLoadBalancer
    Protocol: HTTP
    Port: 80
    DefaultActions:
      - Type: "forward"
        TargetGroupArn: !Ref myTargetGroup
    ListenerAttributes:
```

```
      - Key: "attribute-name"
        Value: "attribute-value"
```

標頭修改屬性

以下是 Application Load Balancer 支援的標頭修改屬性。

`routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name`

修改 X-Amzn-Mtls-Clientcert-Serial-Number 的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_issuer.header_name`

修改 X-Amzn-Mtls-Clientcert-Issuer 的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_subject.header_name`

修改 X-Amzn-Mtls-Clientcert-Subject 的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_validity.header_name`

修改 X-Amzn-Mtls-Clientcert-Validity 的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert_leaf.header_name`

修改 X-Amzn-Mtls-Clientcert-Leaf 的標頭名稱。

`routing.http.request.x_amzn_mtls_clientcert.header_name`

修改 X-Amzn-Mtls-Clientcert 的標頭名稱。

`routing.http.request.x_amzn_tls_version.header_name`

修改 X-Amzn-Tls-Version 的標頭名稱。

`routing.http.request.x_amzn_tls_cipher_suite.header_name`

修改 X-Amzn-Tls-Cipher-Suite 的標頭名稱。

`routing.http.response.server.enabled`

指出是否允許或移除 HTTP 回應伺服器標頭。

`routing.http.response.strict_transport_security.header_value`

新增 Strict-Transport-Security 標頭，以通知瀏覽器應該只使用 HTTPS 存取網站，而且未來任何使用 HTTP 存取網站的嘗試都應該自動轉換為 HTTPS。

`routing.http.response.access_control_allow_origin.header_value`

新增 Access-Control-Allow-Origin 標頭，以指定允許存取伺服器的原始伺服器。

`routing.http.response.access_control_allow_methods.header_value`

新增 Access-Control-Allow-Methods 標頭，以指定從不同原始伺服器存取伺服器時允許哪些 HTTP 方法。

`routing.http.response.access_control_allow_headers.header_value`

新增 Access-Control-Allow-Headers 標頭，以指定跨來源請求期間允許哪些標頭。

`routing.http.response.access_control_allow_credentials.header_value`

新增 Access-Control-Allow-Credentials 標頭，指出瀏覽器是否應在跨來源請求中包含 Cookie 或身分驗證等登入資料。

`routing.http.response.access_control_expose_headers.header_value`

新增 Access-Control-Expose-Headers 標頭，指出瀏覽器可以向請求用戶端公開哪些標頭。

`routing.http.response.access_control_max_age.header_value`

新增 Access-Control-Max-Age 標頭，以秒為單位指定預檢請求結果的快取時間。

`routing.http.response.content_security_policy.header_value`

新增 Content-Security-Policy 標頭來指定瀏覽器強制執行的限制，以協助將特定類型安全威脅的風險降至最低。

`routing.http.response.x_content_type_options.header_value`

新增 X-Content-Type-Options 標頭，以指出是否應遵循 Content-Type 標頭中公告的 MIME 類型，且不得變更。

`routing.http.response.x_frame_options.header_value`

新增 X-Frame-Options 標頭，指出是否允許瀏覽器轉譯影格、iframe、內嵌或物件中的頁面。

刪除 Application Load Balancer 的接聽程式

刪除接聽程式之前，請考慮對應用程式的影響：

- 負載平衡器會立即停止接受接聽程式連接埠上的新連線。
- 作用中連線已關閉。刪除接聽程式時正在進行的任何請求都可能會失敗。

Console

刪除接聽程式

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格上選擇 Load Balancers (負載平衡器)。
3. 選取負載平衡器。
4. 在接聽程式和規則索引標籤上，選取接聽程式的核取方塊，然後選擇管理接聽程式、刪除接聽程式。
5. 出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

AWS CLI

刪除接聽程式

使用 delete-listener 命令。

```
aws elbv2 delete-listener \
    --listener-arn listener-arn
```

Application Load Balancer 的目標群組

目標群組使用您指定的通訊協定和連接埠號碼將請求路由到單獨已註冊的目標，例如 EC2 執行個體。您可以向多個目標群組註冊任一目標。您可以針對每個目標群組設定運作狀態檢查。凡已註冊至負載平衡器的接聽程式規則中指定之目標群組的所有目標，系統將對其執行運作狀態檢查。

每個目標群組會用來將請求轉送到一個或多個註冊的目標。在建立每個接聽程式規則時，您會指定目標群組和條件。規則的條件符合時，會將流量轉送到對應的目標群組。您可以針對不同類型的請求，建立不同的目標群組。例如，針對一般請求建立一個目標群組，然後再針對應用程式微型服務的請求，建立其他的目標群組。每個目標群組只能搭配一個負載平衡器使用。如需詳細資訊，請參閱Application Load Balancer 元件。

您可以針對每個目標群組，指定負載平衡器的運作狀態檢查設定。除非您在建立目標群組時覆寫這些設定，或是在之後修改設定，否則每個目標群組都會使用預設的運作狀態檢查設定。當您在接聽程式的規則中指定目標群組後，負載平衡器會針對自己已啟用可用區域中的目標群組，持續地監控透過該目標群組註冊的所有目標，以了解目標的運作狀態。負載平衡器會將請求路由至運作狀態良好的已註冊目標。

目錄

- 路由組態
- Target type (目標類型)
- IP 地址類型
- 通訊協定版本
- 已登記的目標
- 目標最佳化工具
- 目標群組屬性
- 目標群組運作狀態
- 為您的 Application Load Balancer 建立目標群組
- Application Load Balancer 目標群組的運作狀態檢查
- 編輯 Application Load Balancer 的目標群組屬性
- 向 Application Load Balancer 目標群組註冊目標
- 使用 Lambda 函數作為 Application Load Balancer 的目標
- Application Load Balancer 目標群組的標籤

- 刪除 Application Load Balancer 目標群組

路由組態

根據預設，負載平衡器會使用您在建立目標群組時所指定的通訊協定和埠號，來將請求路由至其目標。或者，您可以在使用目標群組來登錄目標時，覆寫用來將流量轉傳到目標的連接埠。

目標群組支援下列的通訊協定和連接埠：

- Protocols (通訊協定)：HTTP、HTTPS
- Ports (連接埠)：1-65535

使用 HTTPS 通訊協定設定目標群組或使用 HTTPS 運作狀態檢查時，如果任何 HTTPS 接聽程式使用 TLS 1.3 安全政策，則ELBSecurityPolicy-TLS13-1-0-2021-06安全政策將用於目標連線。否則，會使用ELBSecurityPolicy-2016-08安全政策。負載平衡器會使用您在目標上安裝的憑證，與目標建立 TLS 連線。負載平衡器不會驗證這些憑證。因此，您可以使用自我簽署的憑證或已過期的憑證。由於負載平衡器及其目標位於虛擬私有雲端 (VPC) 中，負載平衡器與目標之間的流量會在封包層級進行驗證，因此即使目標上的憑證無效，也不會面臨man-in-the-middle攻擊或詐騙的風險。離開 的流量 AWS 不會有這些相同的保護，而且可能需要其他步驟才能進一步保護流量。

Target type (目標類型)

建立目標群組時，您會指定其目標類型，這會決定您對此目標群組註冊目標時指定的目標類型。在建立目標群組之後，您無法變更其目標類型。

下列是可能的目標類型：

`instance`

以執行個體 ID 來指定目標。

`ip`

目標為 IP 地址。

`lambda`

目標是 Lambda 函數。

如果目標類型是 `ip`，您可以從下列其中一個 CIDR 區塊指定 IP 地址：

- 目標群組 VPC 的子網路
- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

> **Important**
>
> 您無法指定可公開路由傳送的 IP 地址。

所有支援的 CIDR 區塊都可讓您將下列目標註冊至目標群組：

- 與負載平衡器 VPC (相同區域或不同區域) 對等之 VPC 中的執行個體。
- AWS 可透過 IP 地址和連接埠 （ 例如資料庫 ） 定址的 資源。
- 透過 AWS Direct Connect 或 Site-to-Site VPN 連線連結至 的內部部署資源。

> **Note**
>
> 對於在 Local Zone 內部署的 Application Load Balancer，`ip` 目標必須位於相同的 Local Zone，才能接收流量。
> 如需詳細資訊，請參閱什麼是 AWS Local Zones？

如果使用執行個體 ID 來指定目標，會利用在執行個體的主要網路界面中，所指定的主要私有 IP 地址，來將流量轉送到執行個體。如果使用 IP 地址來指定目標，您可以利用來自一個或多個網路界面的任何私有 IP 地址，將流量轉送到執行個體。這可讓執行個體上的多個應用程式，使用相同的連接埠。每個網路界面都可以有自己的安全群組。

如果目標群組的目標類型是 `lambda`，則可以註冊單一 Lambda 函數。當負載平衡器收到 Lambda 函數的請求時，它會呼叫 Lambda 函數。如需詳細資訊，請參閱使用 Lambda 函數作為 Application Load Balancer 的目標。

您可以將 Amazon Elastic Container Service (Amazon ECS) 設定為 Application Load Balancer 的目標。如需詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的使用適用於 Amazon ECS 的 Application Load Balancer。

IP 地址類型

建立新目標群組時，您可以選取目標群組的 IP 地址類型。這會控制用來與目標通訊並檢查目標運作狀態的 IP 版本。

Application Load Balancer 的目標群組支援下列 IP 地址類型：

`ipv4`

負載平衡器會使用 IPv4 與目標通訊。

`ipv6`

負載平衡器會使用 IPv6 與目標通訊。

考量事項

- 負載平衡器會根據目標群組的 IP 地址類型與目標進行通訊。IPv4 目標群組的目標必須接受來自負載平衡器的 IPv4 流量，而 IPv6 目標群組的目標必須接受來自負載平衡器的 IPv6 流量。
- 您無法搭配`ipv4`負載平衡器使用 IPv6 目標群組。
- 您無法向 IPv6 目標群組註冊 Lambda 函數。

通訊協定版本

根據預設，Application Load Balancer 會使用 HTTP/1.1 將請求傳送至目標。您可以使用通訊協定版本，使用 HTTP/2 或 gRPC 將請求傳送至目標。

下表摘要說明請求通訊協定與目標群組通訊協定版本組合的結果。

請求通訊協定	通訊協定版本	結果
HTTP/1.1	HTTP/1.1	成功
HTTP/2	HTTP/1.1	成功
gRPC	HTTP/1.1	錯誤
HTTP/1.1	HTTP/2	錯誤

請求通訊協定	通訊協定版本	結果
HTTP/2	HTTP/2	成功
gRPC	HTTP/2	如果目標支援 gRPC，則成功
HTTP/1.1	gRPC	錯誤
HTTP/2	gRPC	如果是 POST 請求，則成功
gRPC	gRPC	成功

gRPC 通訊協定版本的考量事項

- 唯一支援的接聽程式通訊協定是 HTTPS。
- 接聽程式規則唯一支援的動作類型為 `forward`。
- 支援的目標類型僅為 `instance` 和 `ip`。
- 負載平衡器會剖析 gRPC 請求，並根據套件、服務和方法將 gRPC 呼叫路由至適當的目標群組。
- 負載平衡器支援一元、用戶端串流、伺服器端串流和雙向串流。
- 必須以 `/package.service/method` 格式提供自訂運作狀態檢查方法。
- 必須指定在檢查是否有來自目標的成功回應時要使用的 gRPC 狀態程式碼。
- 您無法使用 Lambda 函數做為目標。

HTTP/2 通訊協定版本的考量事項

- 唯一支援的接聽程式通訊協定是 HTTPS。
- 接聽程式規則唯一支援的動作類型為 `forward`。
- 支援的目標類型僅為 `instance` 和 `ip`。
- 負載平衡器支援一元、用戶端串流、伺服器端串流和雙向串流。每個用戶端 HTTP/2 連線的串流數目上限為 128。

已登記的目標

您的負載平衡器可做為用戶端的單一聯絡窗口，並將傳入的流量分配到各個運作狀態良好的已登錄目標。您可以利用一個或多個群組來登錄每個目標。

如果對應用程式的需求增加，您可以利用一個或多個目標群組來登錄額外的目標，來應付需求。無論設定的閾值為何，只要註冊程序完成且目標通過第一個初始運作狀態檢查，負載平衡器就會開始將流量路由到新註冊的目標。

如果對您應用程式的需求減少，或者您需要為目標提供服務，可以從目標群組取消目標的登錄。取消目標的登錄，會將該目標從目標群組中移除，但不會影響到目標。取消目標的註冊之後，負載平衡器就會立即停止將請求路由到目標。目標會進入 `draining` 狀態，直到處理中的請求已完成。當您準備讓目標再繼續接收請求時，可以將目標註冊到目標群組。

如果是根據執行個體 ID 來註冊目標，您可以使用負載平衡器搭配 Auto Scaling 群組。在將目標群組連接到 Auto Scaling 群組之後，自動擴展會在該群組啟動這些目標時，將目標註冊到目標群組。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling User Guide 中的 Attaching a load balancer to your Auto Scaling group。

限制

- 您無法在相同的 VPC 中註冊另一個 Application Load Balancer 的 IP 地址。如果另一個 Application Load Balancer 位於與負載平衡器 VPC 對等的 VPC 中，您可以註冊其 IP 地址。
- 如果執行個體位於與負載平衡器 VPC（相同區域或不同區域）對等的 VPC 中，則無法依執行個體 ID 註冊執行個體。您可以依照 IP 地址來註冊這些執行個體。

目標最佳化工具

您可以在目標群組上啟用目標最佳化工具。目標最佳化工具可讓您準確強制執行目標的並行請求數量上限。它使用您在目標上安裝和設定的代理程式的協助。若要啟用目標最佳化工具，您可以指定目標群組的目標控制連接埠。此連接埠用於管理代理程式和負載平衡器之間的流量。目標最佳化工具只能在建立目標群組期間啟用。指定的目標控制連接埠無法修改。如需詳細資訊，請參閱the section called “目標最佳化工具”。

目標群組屬性

您可以編輯目標群組的屬性來設定目標群組。如需詳細資訊，請參閱編輯目標群組屬性。

如果目標群組類型為 `instance` 或 `ip`，則支援以下目標群組屬性：

`deregistration_delay.timeout_seconds`

取消註冊目標之前，Elastic Load Balancing 要等待的時間量。範圍介於 0–3600 秒之間。預設值為 300 秒。

`load_balancing.algorithm.type`

路由演算法會決定負載平衡器在路由請求時如何選取目標。值為 `round_robin`、`least_outstanding_requests`或 `weighted_random`。預設值為 `round_robin`。

`load_balancing.algorithm.anomaly_mitigation`

只有在 `load_balancing.algorithm.type`為 時才可用`weighted_random`。指出是否啟用異常緩解。此值為 `on` 或 `off`。預設值為 `off`。

`load_balancing.cross_zone.enabled`

表示是否已啟用跨區域負載平衡。此值為 `true`、`false` 或 `use_load_balancer_configuration`。預設值為 `use_load_balancer_configuration`。

`slow_start.duration_seconds`

時間期間 (秒)，在此期間負載平衡器會將新註冊的目標流量的線性增加共用傳送至目標群組。此範圍介於 30–900 秒之間 (15 分鐘)。預設值為 0 秒 (已停用)。

`stickiness.enabled`

指出是否已啟用黏性工作階段。此值為 `true` 或 `false`。預設值為 `false`。

`stickiness.app_cookie.cookie_name`

應用程式 Cookie 名稱。應用程式 Cookie 名稱不能有下列字首：`AWSALB`、 `AWSALBAPP`或 `AWSALBTG`；它們會保留供負載平衡器使用。

`stickiness.app_cookie.duration_seconds`

應用程式型 Cookie 過期期間 (秒)。在此期間之後，便會將 Cookie 視為過時。最小值為 1 秒，最大值為 7 天 (604800 秒)。預設值為 1 天 (86400 秒)。

`stickiness.lb_cookie.duration_seconds`

持續時間型 Cookie 過期期間 (秒)。在此期間之後，便會將 Cookie 視為過時。最小值為 1 秒，最大值為 7 天 (604800 秒)。預設值為 1 天 (86400 秒)。

`stickiness.type`

黏性的類型。可能的值為 `lb_cookie` 和 `app_cookie`。

`target_group_health.dns_failover.minimum_healthy_targets.count`

運作狀態必須良好的目標數量下限。如果運作狀態良好的目標數量低於此值，請在 DNS 中將節點標示為運作狀態不佳，讓流量只會路由至運作狀態良好的節點。可能的值為 `off`，或從 1 到最高目

標數量的整數。當 時off，DNS 故障停用，這表示即使目標群組中的所有目標都運作狀態不佳，節點也不會從 DNS 中移除。預設為 1。

`target_group_health.dns_failover.minimum_healthy_targets.percentage`

運作狀態必須良好的目標最低百分比。如果運作狀態良好的目標百分比低於此值，請在 DNS 中將節點標記為運作狀態不佳，讓流量只會路由至運作狀態良好的節點。可能的值為 `off`，或介於 1 到 100 之間的整數。當 時off，DNS 故障停用，這表示即使目標群組中的所有目標都運作狀態不佳，節點也不會從 DNS 中移除。預設值為 `off`。

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.count`

運作狀態必須良好的目標最低數量。如果運作狀態良好的目標數量低於此值，請將流量傳送至所有目標，包括運作狀態不佳的目標。範圍介於 1 到目標最高數量。預設為 1。

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage`

運作狀態必須良好的目標最低百分比。如果運作狀態良好的目標百分比低於此值，請將流量傳送至所有目標，包括運作狀態不佳的目標。可能的值為 `off`，或介於 1 到 100 之間的整數。預設值為 `off`。

如果目標群組類型為 `lambda`，則支援以下目標群組屬性：

`lambda.multi_value_headers.enabled`

指出負載平衡器與 Lambda 函數之間的請求和回應標頭交換是否包含值或字串的陣列。可能的值為 `true` 或 `false`。預設值為 `false`。如需詳細資訊，請參閱多值標頭。

目標群組運作狀態

依預設，只要目標群組至少有一個運作狀態良好的目標，就會被視為運作狀態良好。如果您擁有龐大的機群，則只有一個運作狀態良好的目標服務流量是不夠的。相反地，您可以指定必須為運作狀態良好的目標最小計數或百分比，以及當運作狀態良好目標低於指定臨界值時，負載平衡器會採取哪些動作。這可改善應用程式的可用性。

目錄

- 運作運作狀態不佳
- 需求和考量事項
- 監控
- 範例

- 針對您的負載平衡器使用 Route 53 DNS 備援

運作運作狀態不佳

您可以針對下列動作設定運作狀態良好的臨界值：

- DNS 容錯移轉 – 當區域中運作狀態良好的目標低於閾值時，我們會將該區域的負載平衡器節點 IP 地址標記為 DNS 運作狀態不佳。因此，當用戶端解析負載平衡器 DNS 名稱時，流量只會路由至運作狀態良好的區域。
- 路由容錯移轉 – 當區域中運作狀態良好的目標低於閾值時，負載平衡器會將流量傳送至負載平衡器節點可用的所有目標，包括運作狀態不佳的目標。這會增加用戶端連線成功的機會，尤其是當目標暫時無法通過運作狀態檢查時，並降低運作狀態良好目標超載的風險。

需求和考量事項

- 如果您在目標群組上啟用目標最佳化工具，建議您將目標群組的運作狀態檢查連接埠設定為與 TARGET_CONTROL_DATA_ADDRESS 中的連接埠相同。這可確保如果代理程式運作狀態不佳，目標將無法通過運作狀態檢查。如需詳細資訊，請參閱the section called “目標最佳化工具”。
- 您無法在目標為 Lambda 函數的目標群組中使用此功能。如果 Application Load Balancer 是 Network Load Balancer 或 Global Accelerator 的目標，請勿設定 DNS 備援的閾值。
- 如果您為動作指定兩種類型的閾值 (計數和百分比)，則當違反任一閾值時，負載平衡器都會採取動作。
- 如果您指定這兩個動作的臨界值，DNS 備援的臨界值必須大於或等於路由容錯移轉的臨界值，以便 DNS 備援發生在路由容錯移轉或之前。
- 如果您將臨界值指定為百分比，我們會根據向目標群組註冊的目標總數來動態計算值。
- 目標總數取決於是關閉還是開啟跨區域負載平衡。如果關閉跨區域負載平衡，則每個節點只會將流量傳送到其自身區域中的目標，這代表臨界值會分別套用至每個已啟用區域中的目標數目。如果開啟跨區域負載平衡，則每個節點會將流量傳送到所有已啟用區域中的所有目標，這代表指定的臨界值會套用至所有已啟用區域中的目標總數。如需詳細資訊，請參閱跨區域負載平衡。
- 發生 DNS 容錯移轉時，會影響與負載平衡器相關聯的所有目標群組。確保剩餘區域中有足夠的容量來處理這些額外的流量，尤其是在跨區域負載平衡關閉的情況下。
- 使用 DNS 容錯移轉時，我們會將運作狀態不佳區域的 IP 地址從負載平衡器的 DNS 主機名稱中移除。不過，本機用戶端 DNS 快取可能會包含這些 IP 地址，直到 DNS 記錄中的存活期 (TTL) 到期 (60 秒) 為止。

- 使用 DNS 容錯移轉時，如果有多個目標群組連接到 Application Load Balancer，且一個目標群組在區域中運作狀態不佳，則 DNS 運作狀態檢查會成功，如果該區域中至少有一個其他目標群組運作狀態良好。
- 使用 DNS 備援時，如果將所有負載平衡器區域視為運作狀態不佳，負載平衡器會將流量傳送到所有區域，包括運作狀態不佳的區域。
- 除了是否有足夠運作狀態良好的目標可能導致 DNS 備援之外，還有其他因素，例如區域的運作狀況。

監控

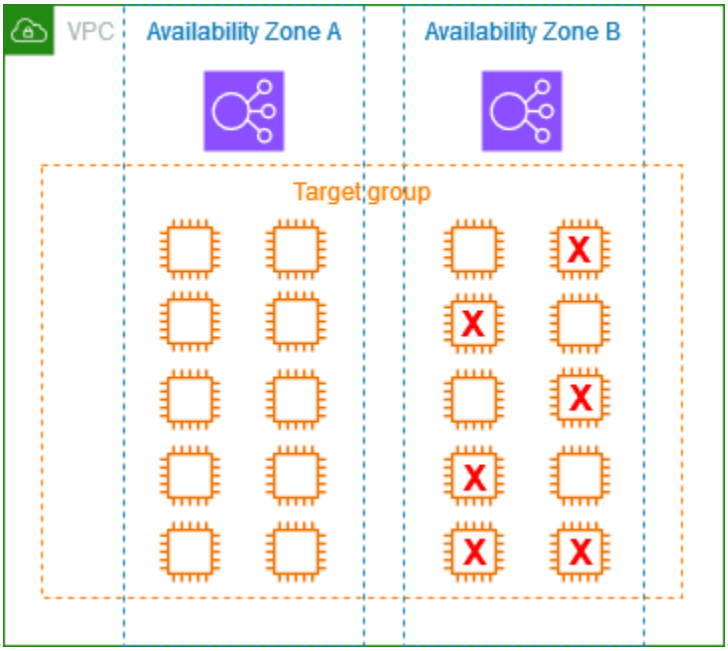
若要監控目標群組的運作狀態，請參閱目標群組運作狀態的 CloudWatch 指標。

範例

以下範例示範如何套用目標群組運作狀態設定。

案例

- 支援 A 和 B 兩個可用區域的負載平衡器
- 每個可用區域包含 10 個已註冊目標
- 目標群組具有下列目標群組運作狀態設定：
  - DNS 備援 - 50%
  - 路由容錯移轉 - 50%
- 可用區域 B 中有六個目標失敗



如果停用跨區域負載平衡

- 每個可用區域中的負載平衡器節點只能將流量傳送到其可用區域中的 10 個目標。
- 可用區域 A 中有 10 個運作狀態良好的目標，符合運作狀態目標的必要百分比。負載平衡器會繼續在 10 個運作狀態良好的目標之間分配流量。
- 可用區域 B 中只有 4 個運作狀態良好的目標，這是可用區域 B 中負載平衡器節點目標的 40%，因為小於運作狀態良好目標的必要百分比，所以負載平衡器會採取下列動作：
  - DNS 備援 - 可用性區域 B 在 DNS 中標示為運作狀態不良。由於用戶端無法將負載平衡器名稱解析為可用區域 B 中的負載平衡器節點，且可用區域 A 運作狀態良好，因此用戶端會將新的連線傳送至可用區域 A。
  - 路由容錯移轉 - 當新連線明確傳送至可用區域 B 時，負載平衡器會將流量分配給可用性區域 B 中的所有目標，包括運作狀態不佳的目標。這樣可以防止剩餘運作狀態良好的目標中斷。

如果啟用跨區域負載平衡

- 每個負載平衡器節點都可以將流量傳送到兩個可用區域的所有 20 個已註冊目標。
- 可用區域 A 中有 10 個運作狀態良好的目標，而可用區域 B 中有 4 個運作狀態良好的目標，總共有 14 個運作狀態良好目標。這是兩個可用區域中負載平衡器節點目標的 70%，符合運作狀態良好目標的必要百分比。
- 負載平衡器會在兩個可用區域中 14 個運作狀況良好的目標之間分配流量。

針對您的負載平衡器使用 Route 53 DNS 備援

如果您使用 Route 53 將 DNS 查詢路由傳送到負載平衡器，您也可以使用 Route 53 設定負載平衡器的 DNS 備援。在容錯移轉組態中，Route 53 會檢查負載平衡器的目標群組目標的運作狀態，以判斷是否可用。如果沒有負載平衡器註冊的狀態良好目標，或者負載平衡器本身運作狀態不佳，Route 53 會將流量路由到另一可用資源，例如運作狀態良好的負載平衡器或 Amazon S3 中的靜態網站。

例如，假設您有一個 `www.example.com` Web 應用程式，而且您需要在後方執行兩個負載平衡器備援執行個體，位於不同的區域。您希望流量在一個區域主要路由到負載平衡器，而且您想要在其他區域使用負載平衡器，以供失敗時備份。如果您設定 DNS 容錯移轉，您可以指定您的主要和次要 (備份) 負載平衡器。Route 53 會引導流量到可用的主要負載平衡器，或是次要負載平衡器。

評估目標運作狀態的運作方式

- 如果 Application Load Balancer 的別名記錄`Yes`上的評估目標運作狀態設為 ，Route 53 會評估`alias target`值所指定資源的運作狀態。Route 53 使用目標群組運作狀態檢查。
- 如果連接至 Application Load Balancer 的所有目標群組都正常運作，Route 53 會將別名記錄標記為正常運作。如果您已設定目標群組的閾值，且符合其閾值，則會通過運作狀態檢查。否則，如果目標群組包含至少一個運作狀態良好的目標，則會通過運作狀態檢查。如果運作狀態檢查通過，Route 53 會根據您的路由政策傳回記錄。如果使用容錯移轉路由政策，Route 53 會傳回主要記錄。
- 如果連接到 Application Load Balancer 的任何目標群組運作狀態不佳，別名記錄會失敗 Route 53 運作狀態檢查 （故障開啟）。如果使用 評估目標運作狀態，容錯移轉路由政策會將流量重新導向至次要資源。
- 如果連接到 Application Load Balancer 的所有目標群組都是空的 （沒有目標），Route 53 會將記錄視為運作狀態不佳 （故障開啟）。如果使用 評估目標運作狀態，容錯移轉路由政策會將流量重新導向至次要資源。

如需詳細資訊，請參閱《Amazon Route 53 開發人員指南》中的使用負載平衡器目標群組運作狀態閾值來改善部落格中的可用性和設定 DNS 容錯移轉。 AWS

為您的 Application Load Balancer 建立目標群組

您會向目標群組註冊您的目標。根據預設，負載平衡器會使用您針對目標群組所指定的埠號和通訊協定，來將請求傳送到登錄的目標。在透過目標群組來註冊每個目標時，您可以覆寫此埠號。

在建立目標群組之後，您可以新增標籤。

若要將流量路由到目標群組中的目標，請在建立接聽程式或為接聽程式建立規則時，於動作中指定目標群組。如需詳細資訊，請參閱Application Load Balancer 的接聽程式規則。您可以在多個接聽程式中指定相同的目標群組，但這些接聽程式必須屬於相同的 Application Load Balancer。若要將目標群組與負載平衡器搭配使用，您必須確認任何其他負載平衡器的接聽程式未使用此目標群組。

您可以隨時從目標群組新增或移除目標。如需詳細資訊，請參閱向 Application Load Balancer 目標群組註冊目標。您也可以修改目標群組的運作狀態檢查設定。如需詳細資訊，請參閱更新 Application Load Balancer 目標群組的運作狀態檢查設定。

Console

若要建立目標群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇 Create target group (建立目標群組)。
4. 在選擇目標類型中，選取執行個體可依執行個體 ID 註冊目標，選取 IP 地址可依 IP 地址註冊目標，或選取 Lambda 函數 依 Lambda 函數註冊目標。
5. 針對 Target group name (目標群組名稱)，輸入目標群組的名稱。此名稱在每個帳戶的每個區域中都必須是唯一的，其長度上限為 32 個字元，並且必須僅包含英數字元或連字號，且開頭或結尾不可以是連字號。
6. (選用) 針對 Protocol (通訊協定) 和 Port (連接埠)，視需要修改預設值。
7. 如果目標類型為執行個體或 IP 地址，請選擇 IPv4 或 IPv6 作為 IP 地址類型，否則請跳至下一個步驟。

   請注意，只有具有所選 IP 地址類型的目標才能包含在此目標群組中。建立目標群組後，便無法變更 IP 地址類型。
8. 針對 VPC (VPC) 選擇虛擬私有雲端 (VPC)。請注意，對於 IP addresses (IP 地址) 目標類型，可供選擇的 VPC 是支援您在上一個步驟中選擇之 IP address type (IP 地址類型) 的 VPC。
9. (選用) 針對 Protocol version (通訊協定版本)，視需要修改預設值。如需詳細資訊，請參閱the section called “通訊協定版本”。
10. (選用) 在 Health checks (運作狀態檢查) 區段中，視需要修改預設設定。如需詳細資訊，請參閱the section called “運作狀態檢查設定”。
11. 如果目標類型為 Lambda 函數，您可以透過選取 Health checks (運作狀態檢查) 區段中的 啟用 (Enable) 來啟用運作狀態檢查。

12. （選用）若要在目標群組上啟用目標最佳化工具，請指定目標控制連接埠。目標群組建立後，無法修改連接埠。目標最佳化工具可搭配您在目標上安裝的 代理程式來運作。如需詳細資訊，請參閱the section called “目標最佳化工具”。
13. (選用) 新增一個或多個標籤，如下所示：
    a. 展開 Tags (標籤) 區段。
    b. 選擇 Add tag (新增標籤)。
    c. 輸入標籤金鑰和標籤值。
14. 選擇下一步。
15. (選用) 新增一個或多個目標，如下所示：
    - 如果目標類型為執行個體，請選取一個或多個執行個體，輸入一個或多個連接埠，然後選擇包含為下方待處理項目。

      注意：執行個體必須具有指派的主要 IPv6 地址，才能在 IPv6 目標群組中註冊。
    - 如果目標類型是 IP 地址，請執行下列動作：
      a. 從清單中選取網路 VPC，或選擇其他私人 IP 地址。
      b. 手動輸入 IP 地址，或使用執行個體詳細資料尋找 IP 地址。一次最多可輸入五個 IP 地址。
      c. 輸入用於將流量路由到指定 IP 地址的連接埠。
      d. 選擇包含為下方待處理項目。
    - 如果目標類型是 Lambda 函數，請指定單一 Lambda 函數，或省略此步驟，稍後再指定 Lambda 函數。
16. 選擇 Create target group (建立目標群組)。

AWS CLI

若要建立目標群組

使用 create-target-group 命令。下列範例會使用 HTTP 通訊協定、IP 地址註冊的目標、一個標籤和預設運作狀態檢查設定來建立目標群組。

```
aws elbv2 create-target-group \
    --name my-target-group \
    --protocol HTTP \
    --port 80 \
    --target-type ip \
```

```
    --vpc-id vpc-1234567890abcdef0 \
    --tags Key=department,Value=123
```

註冊目標

使用 register-targets 命令向目標群組註冊目標。如需範例，請參閱 the section called “登記目標”。

CloudFormation

若要建立目標群組

定義 AWS::ElasticLoadBalancingV2::TargetGroup 類型的資源。下列範例會建立目標群組，其中包含 HTTP 通訊協定、依 IP 地址註冊的目標、一個標籤、預設運作狀態檢查設定，以及兩個已註冊的目標。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: 10.0.50.10
          Port: 80
        - Id: 10.0.50.20
          Port: 80
```

Application Load Balancer 目標群組的運作狀態檢查

Application Load Balancer 會定期將請求傳送到已註冊的目標來測試其狀態。這些測試稱為運作狀況檢查。

每個負載平衡器節點只會將請求路由至負載平衡器已啟用可用區域內運作狀態良好的目標。每個負載平衡器節點會使用各目標註冊所屬目標群組的運作狀態檢查設定，檢查該目標的運作狀態。目標註冊後，必須通過一次運作狀態檢查，才算運作狀態良好。每次運作狀態檢查完成後，負載平衡器節點即會關閉其為執行運作狀態檢查而建立的連線。

如果目標群組只包含運作狀態不佳的已註冊目標，負載平衡器會將請求路由至所有這些目標，而不論這些目標的運作狀態為何。這表示如果所有已啟用可用區域中的所有目標同時都未通過運作狀態檢查，則負載平衡器會故障開啟。故障開啟的結果是系統會根據負載平衡演算法，允許流量傳輸到所有已啟用可用區域中的所有目標，無論這些目標的運作狀態為何。

運作狀態檢查不支援 WebSocket。

如需詳細資訊，請參閱the section called “目標群組運作狀態”。

您可以使用運作狀態檢查日誌來擷取對負載平衡器已註冊目標執行的運作狀態檢查詳細資訊，並將其儲存為 Amazon S3 中的日誌檔案。您可以使用這些運作狀態檢查日誌來疑難排解目標的問題。如需詳細資訊，請參閱運作狀態檢查日誌。

目錄

- 運作狀態檢查設定
- 目標運作狀態
- 運作狀態檢查原因代碼
- 檢查 Application Load Balancer 目標的運作狀態
- 更新 Application Load Balancer 目標群組的運作狀態檢查設定

運作狀態檢查設定

您需要按下表中的描述為目標群組中的目標設定運作狀態檢查。表中使用的設定名稱是 API 中使用的名稱。負載平衡器會使用指定的連接埠、通訊協定和運作狀態檢查路徑，每隔 HealthCheckIntervalSeconds 秒將運作狀態檢查請求傳送至每個已註冊的目標。每個運作狀態檢查請求是各自獨立，且在整個間隔內持續保持此結果。目標回應所花的時間不影響下次運作狀態檢查請求的間隔。如果運作狀態檢查連續失敗超過 UnhealthyThresholdCount 次，負載平衡器會停用該目標。當運作狀態檢查連續成功超過 HealthyThresholdCount 次時，負載平衡器重新啟用該目標。

請注意，當您取消註冊目標時，這會減少 HealthyHostCount，但不會增加 UnhealthyHostCount。

設定	Description
HealthCheckProtocol	負載平衡器對目標執行運作狀態檢查時使用的通訊協定。對於 Application Load Balancer，可能的通訊協定為 HTTP 和 HTTPS。預設為 HTTP 通訊協定。

設定	Description
	這些通訊協定會使用 HTTP GET 方法，來傳送運作狀態檢查請求。
HealthCheckPort	負載平衡器對目標執行運作狀態檢查時使用的連接埠。預設為使用每個目標從負載平衡器接收流量的連接埠。
HealthCheckPath	目標上運作狀態檢查的目的地。 如果通訊協定版本是 HTTP/1.1 或 HTTP/2，請指定有效的 URI (/path?query)。預設為 /. 如果通訊協定版本是 gRPC，則使用 `/package.service/method` 格式指定自訂運作狀態檢查方法的路徑。預設值為 `/AWS.ALB/healthcheck` 。
HealthCheckTimeoutSeconds	以秒為單位的時間量，若目標在此期間內毫無回應即表示運作狀態檢查失敗。範圍介於 2 到 120 秒之間。如果目標類型是 `instance` 或 `ip`，則預設為 5 秒，如果是 `lambda`，則預設為 30 秒。
HealthCheckIntervalSeconds	個別目標每次執行運作狀態檢查的大約間隔時間量，以秒為單位。範圍介於 5–300 秒之間。如果目標類型是 `instance` 或 `ip`，則預設為 30 秒，如果是 `lambda`，則預設為 35 秒。
HealthyThresholdCount	將運作狀態不佳的目標視為運作狀態良好之前，運作狀態檢查需連續成功的次數。範圍介於 2–10 之間。預設值為 5。
UnhealthyThresholdCount	將目標視為運作狀態不佳之前，運作狀態檢查需連續失敗的次數。範圍介於 2–10 之間。預設為 2。

設定	Description
Matcher	檢查是否收到來自目標的成功回應時所使用的代碼。這些在主控台中稱為成功代碼。 如果通訊協定版本是 HTTP/1.1 或 HTTP/2，則值範圍是 200 到 499。您可以指定多個值 (例如，"200,202") 或值範圍 (例如，"200-299")。預設值為 200。 如果通訊協定版本是 gRPC，則值範圍是 0 到 99。您可以指定多個值 (例如，"0,1") 或值範圍 (例如，"0-5")。預設值為 12。

目標運作狀態

在負載平衡器向目標傳送運作狀態檢查請求之前，您必須向目標群組註冊該目標，由接聽程式規則中指定其目標群組，並確保負載平衡器已啟用該目標的可用區域。目標必須通過初次運作狀態檢查，才能從負載平衡器收到請求。在目標通過初次運作狀態檢查後，它的狀態是 Healthy。

下表說明已註冊目標的運作狀態可能的值。

Value	Description
initial	負載平衡器正在註冊目標或對目標執行初始運作狀態檢查。 相關原因代碼： Elb.RegistrationInProgress | Elb.InitialHealthChecking
healthy	目標的運作狀態良好。 相關原因代碼：無
unhealthy	目標未回應運作狀態檢查或未通過運作狀態檢查。

Value	Description
	相關原因碼：Target.ResponseCodeMismatch | Target.Timeout | Target.FailedHealthChecks | Elb.InternalError
unused	目標未向目標群組註冊、未在接聽程式規則中使用目標群組、目標位於未啟用的可用區域，或目標處於已停止或已終止狀態。 相關原因碼：Target.NotRegistered | Target.NotInUse | Target.InvalidState | Target.IpUnusable
draining	目標正在取消註冊，連接耗盡作業進行中。 相關原因碼：Target.DeregistrationInProgress
unavailable	目標群組的運作狀態檢查已停用。 相關原因碼：Target.HealthCheckDisabled

運作狀態檢查原因代碼

如果目標的狀態是 Healthy 以外的任何值，API 將傳回問題的原因代碼和描述，而且主控台會顯示同樣的描述。開頭為 Elb 的原因代碼源自負載平衡器端，而開頭為 Target 的原因代碼源自目標端。如需運作狀態檢查失敗可能原因的詳細資訊，請參閱 Troubleshooting。

原因代碼	Description
Elb.InitialHealthChecking	初始運作狀態檢查正進行中
Elb.InternalError	運作狀態檢查由於內部錯誤而失敗
Elb.RegistrationInProgress	目標註冊正進行中

原因代碼	Description
`Target.DeregistrationInProgress`	目標取消註冊正進行中
`Target.FailedHealthChecks`	運作狀態檢查失敗
`Target.HealthCheckDisabled`	運作狀態檢查已停用
`Target.InvalidState`	目標處於停止狀態 目標處於終止狀態 目標處於終止或停止狀態 目標處於無效狀態
`Target.IpUnusable`	IP 地址不能做為目標，因為負載平衡器正在使用它
`Target.NotInUse`	目標群組未設定為接收來自負載平衡器的流量 目標位於負載平衡器未啟用的可用區域
`Target.NotRegistered`	目標未向目標群組註冊
`Target.ResponseCodeMismatch`	運作狀態檢查失敗，顯示以下代碼：[code]
`Target.Timeout`	請求逾時

檢查 Application Load Balancer 目標的運作狀態

您可以檢查已向目標群組註冊的各個目標的運作狀態。如需運作狀態檢查失敗的說明，請參閱故障診斷：已註冊的目標不在服務中。

您可以使用運作狀態檢查日誌來擷取對負載平衡器已註冊目標執行的運作狀態檢查詳細資訊，並將其儲存為 Amazon S3 中的日誌檔案。您可以使用這些運作狀態檢查日誌來疑難排解目標的問題。如需詳細資訊，請參閱運作狀態檢查日誌。

Console

檢查目標的運作狀態

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 詳細資訊索引標籤會顯示目標總數，以及每個運作狀態的目標數量。
5. 在 Targets (目標) 標籤上，Status (狀態) 欄指出各目標的狀態。
6. 如果狀態是 `Healthy` 以外的任何值，則狀態詳細資料資料欄會包含更多資訊。

接收有關狀態不良目標的電子郵件通知

使用 CloudWatch 警示來觸發 Lambda 函數，以傳送運作狀態不佳目標的詳細資料。如需逐步指示，請參閱下列部落格文章：Identifying unhealthy targets of your load balancer (識別負載平衡器狀態不良的目標)。

AWS CLI

檢查目標的運作狀態

使用 describe-target-health 命令。此範例會篩選輸出，只包含運作狀態不佳的目標。對於運作狀態不佳的目標，輸出會包含原因代碼。

```
aws elbv2 describe-target-health \
    --target-group-arn target-group-arn \
    --query "TargetHealthDescriptions[?TargetHealth.State!='healthy'].
[Target.Id,TargetHealth.State,TargetHealth.Reason]" \
    --output table
```

以下為範例輸出。

```
-----------------------------------------------
|            DescribeTargetHealth             |
+--------------+---------+--------------------+
|  172.31.0.57 |  unused |  Target.NotInUse   |
|  172.31.0.50 |  unused |  Target.NotInUse   |
+--------------+---------+--------------------+
```

目標狀態和原因代碼

下列清單顯示每個目標狀態的可能原因代碼。

目標狀態為 healthy

未提供原因代碼。

目標狀態為 initial

- `Elb.RegistrationInProgress` - 目標正在向負載平衡器註冊。
- `Elb.InitialHealthChecking` - 負載平衡器仍在向目標傳送判斷其運作狀態所需的最低運作狀態檢查次數。

目標狀態為 unhealthy

- `Target.ResponseCodeMismatch` - 運作狀態檢查未傳回預期的 HTTP 程式碼。
- `Target.Timeout` - 運作狀態檢查請求逾時。
- `Target.FailedHealthChecks` - 負載平衡器在建立與目標的連線或目標回應格式錯誤時收到錯誤。
- `Elb.InternalError` - 運作狀態檢查因內部錯誤而失敗。

目標狀態為 unused

- `Target.NotRegistered` - 目標未向目標群組註冊。
- `Target.NotInUse` - 目標群組不會被任何負載平衡器使用，或目標位於未為其負載平衡器啟用的可用區域中。
- `Target.InvalidState` - 目標處於已停止或終止狀態。
- `Target.IpUnusable` - 目標 IP 地址保留供負載平衡器使用。

目標狀態為 draining

- `Target.DeregistrationInProgress` - 目標正在進行取消註冊，且取消註冊延遲期間尚未過期。

目標狀態為 unavailable

- `Target.HealthCheckDisabled` - 目標群組的運作狀態檢查已停用。

更新 Application Load Balancer 目標群組的運作狀態檢查設定

您可以隨時更新目標群組的運作狀態檢查設定。如需運作狀態檢查設定清單，請參閱 the section called “運作狀態檢查設定”。

Console

更新運作狀態檢查設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在 Health checks (運作狀態檢查) 標籤上，選擇 Edit (編輯)。
5. 在編輯運作狀態檢查設定頁面上，視需要修改設定。
6. 選擇儲存變更。

AWS CLI

更新運作狀態檢查設定

使用 modify-target-group 命令。下列範例會更新 HealthyThresholdCount 和 HealthCheckTimeoutSeconds 設定。

```
aws elbv2 modify-target-group \
    --target-group-arn target-group-arn \
    --healthy-threshold-count 3 \
    --health-check-timeout-seconds 20
```

CloudFormation

更新運作狀態檢查設定

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源，以包含更新的運作狀態檢查設定。下列範例會更新 HealthyThresholdCount 和 HealthCheckTimeoutSeconds 設定。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      HealthyThresholdCount: 3
```

```
    HealthCheckTimeoutSeconds: 20
```

編輯 Application Load Balancer 的目標群組屬性

為 Application Load Balancer 建立目標群組之後，您可以編輯其目標群組屬性。

目標群組屬性

- 取消登記的延遲
- 路由演算法
- 慢速啟動模式
- 運作狀態設定
- 跨區域負載平衡
- 自動目標權重 (ATW)
- 黏性工作階段

取消登記的延遲

Elastic Load Balancing 會停止將請求傳送給正在取消註冊的目標。在預設情況下，Elastic Load Balancing 需要等待 300 秒，才能完成取消註冊程序，這有助於至目標的傳輸中請求完成。若要變更 Elastic Load Balancing 等待的時間，請更新取消註冊延遲時間值。

取消註冊中的目標，其初始狀態為 `draining`。經過取消註冊延遲之後，取消註冊程序會完成，並且目標的狀態為 `unused`。如果目標是 Auto Scaling 群組的一部分，可加以終止和取代。

如果取消註冊的目標沒有傳輸中的請求，也沒有作用中連線，Elastic Load Balancing 會立即完成取消註冊程序，而不會等候取消註冊延遲時間結束。不過，即使目標取消註冊完成，目標的狀態仍會顯示為 `draining`，直到取消註冊延遲逾時結束。逾時結束後，目標會轉變為 `unused` 狀態。

如果取消註冊目標在取消註冊延遲經過之前終止連線，用戶端會收到 500 層級的錯誤回應。

Console

更新取消註冊延遲值

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。

4. 在屬性索引標籤中，選擇編輯。
5. 在目標取消註冊管理窗格中，輸入取消註冊延遲的新值。
6. 選擇儲存變更。

AWS CLI

更新取消註冊延遲值

使用 modify-target-group-attributes 命令搭配 `deregistration_delay.timeout_seconds` 屬性。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=deregistration_delay.timeout_seconds,Value=60"
```

CloudFormation

更新取消註冊延遲值

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 `deregistration_delay.timeout_seconds` 屬性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "deregistration_delay.timeout_seconds"
          Value: "60"
```

路由演算法

路由演算法是負載平衡器在判斷哪些目標將接收請求時使用的方法。預設會使用循環配置路由演算法，在目標群組層級路由請求。根據您的應用程式需求，也可以使用最不未完成的請求和加權隨機路由演算法。目標群組一次只能有一個作用中的路由演算法，但路由演算法可以視需要更新。

如果您啟用黏性工作階段，選取的路由演算法會用於初始目標選擇。來自相同用戶端的未來請求將轉送至相同的目標，繞過選取的路由演算法。如果您已啟用目標最佳化工具，則路由演算法只能是循環配置。

循環配置

- 循環配置路由演算法會依序將請求平均路由到目標群組中運作狀態良好的目標。
- 當收到的請求在複雜性方面類似、已註冊的目標在處理能力方面類似，或者您需要在目標之間平均分配請求時，通常會使用此演算法。

最少未完成的請求

- 最不未完成的請求路由演算法會將請求路由到正在進行中請求數量最低的目標。
- 當收到的請求的複雜性不同時，通常會使用此演算法，註冊的目標的處理能力也不同。
- 當支援 HTTP/2 的負載平衡器使用僅支援 HTTP/1.1 的目標時，會將請求轉換為多個 HTTP/1.1 請求。在此組態中，最少未完成的請求演算法會將每個 HTTP/2 請求視為多個請求。
- 使用 WebSockets 時，會使用最不未完成的請求演算法來選取目標。選取目標之後，負載平衡器會建立與目標的連線，並透過此連線傳送所有訊息。
- 最少未完成的請求路由演算法無法與慢速啟動模式搭配使用。

加權隨機

- 加權隨機路由演算法會以隨機順序，將請求平均路由到目標群組中運作狀態良好的目標。
- 此演算法支援自動目標權重 (ATW) 異常緩解。
- 加權隨機路由演算法無法與慢速啟動模式搭配使用。
- 加權隨機路由演算法無法與黏性工作階段搭配使用。

Console

更新路由演算法

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。

5. 在流量組態窗格中，針對負載平衡演算法，選擇循環配置、最少未完成的請求或加權隨機。
6. 選擇儲存變更。

AWS CLI

更新路由演算法

使用 modify-target-group-attributes 命令搭配 `load_balancing.algorithm.type` 屬性。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes
 "Key=load_balancing.algorithm.type,Value=least_outstanding_requests"
```

CloudFormation

更新路由演算法

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 `load_balancing.algorithm.type` 屬性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.algorithm.type"
          Value: "least_outstanding_requests"
```

慢速啟動模式

在預設情況下，在向目標群組註冊目標並通過初始運作狀態檢查之後，目標隨即會開始接收其請求的完整共用。使用慢速啟動模式可讓目標在負載平衡器向它們傳送請求的完整共用之前，有時間進行暖機。

啟用目標群組的慢啟動後，當目標群組被視為運作狀態良好時，其目標會進入慢速啟動模式。慢速啟動模式的目標，會在設定的慢啟動超過持續的時間或目標變得狀態不良時，結束慢速啟動模式。負載平衡器會線性增加可以在慢速啟動模式中傳送到目標的請求數量。在運作狀態良好的目標退出慢速啟動模式之後，負載平衡器可以將完整份額的請求傳送給它。

考量事項

- 啟用目標群組的慢啟動時，運作狀態良好的已註冊目標不會進入慢速啟動模式。
- 為空白目標群組啟用慢速啟動，然後使用單一註冊操作註冊目標時，這些目標不會進入慢速啟動模式。只有在至少一個運作狀態良好的目標不處於慢速啟動模式時，新註冊的目標才會進入慢速啟動模式。
- 如果您將處於慢速啟動模式的目標取消註冊，該目標會退出慢速啟動模式。如果您再次註冊相同的目標，當目標群組視為運作狀態良好時，它會進入慢速啟動模式。
- 如果處於慢速啟動模式的目標變得運作狀態不佳，目標就會結束慢速啟動模式。當目標狀況良好時，它會再次進入慢速啟動模式。
- 使用最不未完成的請求或加權隨機路由演算法時，您無法啟用慢速啟動模式。

Console

更新慢速啟動持續時間值

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 在流量組態窗格中，輸入慢速啟動持續時間的新值。若要停用慢速啟動模式，請輸入 0。
6. 選擇儲存變更。

AWS CLI

更新慢速啟動持續時間值

使用 modify-target-group-attributes 命令搭配 `slow_start.duration_seconds` 屬性。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
```

```
    --attributes "Key=slow_start.duration_seconds,Value=30"
```

CloudFormation

更新慢速啟動持續時間值

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 `slow_start.duration_seconds` 屬性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "slow_start.duration_seconds"
          Value: "30"
```

運作狀態設定

根據預設，Application Load Balancer 會監控目標的運作狀態，並將請求路由至運作狀態良好的目標。不過，如果負載平衡器沒有足夠的運作狀態良好的目標，它會自動將流量傳送至所有已註冊的目標（失敗開啟）。您可以修改目標群組的目標群組運作狀態設定，以定義 DNS 容錯移轉和路由容錯移轉的閾值。如需詳細資訊，請參閱the section called “目標群組運作狀態”。

Console

修改目標群組運作狀態設定

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的負載平衡中，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 檢查是否開啟或關閉跨區域負載平衡。視需要更新此設定，以確保您有足夠的容量可在區域發生故障時處理額外的流量。

6. 展開目標群組運作狀況需求。
7. 針對組態類型，建議您選擇統一組態，這兩個動作都會設定相同的臨界值。
8. 對於狀態良好的狀態要求，請執行下列其中一項：
   - 選擇最小運作狀況目標計數，然後輸入從 1 到目標群組目標數目上限的數字。
   - 選擇最小狀態良好目標百分比，然後輸入 1 到 100 之間的數字。
9. 選擇儲存變更。

AWS CLI

修改目標群組運作狀態設定

使用 modify-target-group-attributes 指令。下列範例會將兩個運作狀態不佳的動作的運作狀態良好閾值設定為 50%。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \

 "Key=target_group_health.dns_failover.minimum_healthy_targets.percentage,Value=50"
 \

 "Key=target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage,Value=5
```

CloudFormation

修改目標群組運作狀態設定

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源。下列範例會將兩個運作狀態不佳的動作的運作狀態良好閾值設定為 50%。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
```

```
        - Key: "target_group_health.dns_failover.minimum_healthy_targets.percentage"
          Value: "50"
        - Key:
"target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage"
          Value: "50"
```

跨區域負載平衡

負載平衡器的節點會將請求從用戶端分發到已註冊的目標。跨區域負載平衡開啟後，每個負載平衡器節點會將流量分散到所有已註冊可用區域內的已註冊目標。跨區域負載平衡關閉後，每個負載平衡器節點只會將流量分散到其可用區域內已註冊的目標。如果區域故障網域優先於地區故障網域，就可能發生此情形，以確保運作狀態良好的區域不受運作狀態不佳區域的影響，也可能是為改善整體延遲。

使用 Application Load Balancer 時，跨區域負載平衡一律會在負載平衡器層級開啟，而且無法關閉。對於目標群組，預設設定為使用負載平衡器設定，但您可以在目標群組層級明確關閉跨區域負載平衡來覆寫預設設定。

考量事項

- 跨區域負載平衡關閉後，不支援目標粘性。
- 跨區域負載平衡關閉後，不支援將 Lambda 函數作為目標。
- 如果有任何目標的參數 `AvailabilityZone` 設定為 `all`，則嘗試透過 `ModifyTargetGroupAttributes` API 關閉跨區域負載平衡會導致錯誤。
- 註冊目標時需要 `AvailabilityZone` 參數。跨區域負載平衡關閉後，僅允許特定可用區域值。否則，系統會忽略此參數並將其當作 `all`。

最佳實務

- 針對您預期使用的所有可用區域，為每個目標群組規劃足夠的目標容量。如果無法為所有參與的可用區域規劃足夠容量，建議將跨區域負載平衡保持開啟的狀態。
- 如果 Application Load Balancer 設定有多個目標群組，請確定所有目標群組都在設定的區域內參與相同的可用區域。這是為了避免跨區域負載平衡關閉後可用區域變成空白的狀態，因為這會對進入空白可用區域的所有 HTTP 請求觸發 503 錯誤。
- 避免建立空白子網路。Application Load Balancer 會透過 DNS 公開空白子網路的區域 IP 地址，這會針對 HTTP 請求觸發 503 錯誤。
- 還可能會發生這種情況：關閉跨區域負載平衡的目標群組為每個可用區域都計劃有足夠的目標容量，但可用區域中的所有目標都變成運作狀態不佳。當至少一個目標群組包含的所有目標都運作狀態不佳

時，就會將負載平衡器節點的 IP 地址從 DNS 移除。當目標群組具有至少一個運作狀態良好的目標之後，IP 地址就會還原到 DNS。

關閉跨區域負載平衡

您可以隨時為 Application Load Balancer 目標群組關閉跨區域負載平衡。

Console

關閉跨區域負載平衡

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤上，選取編輯。
5. 在目標選取組態窗格中，選擇關閉以進行跨區域負載平衡。
6. 選擇儲存變更。

AWS CLI

關閉跨區域負載平衡

使用 modify-target-group-attributes 命令並將 `load_balancing.cross_zone.enabled` 屬性設為 `false`。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=load_balancing.cross_zone.enabled,Value=false"
```

CloudFormation

關閉跨區域負載平衡

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 `load_balancing.cross_zone.enabled` 屬性。

```
Resources:
```

```
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.cross_zone.enabled"
          Value: "false"
```

開啟跨區域負載平衡

您可以隨時為 Application Load Balancer 目標群組開啟跨區域負載平衡。目標群組層級的跨區域負載平衡設定會覆寫負載平衡器層級的設定。

Console

關閉跨區域負載平衡

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤上，選取編輯。
5. 在目標選取組態窗格中，選擇開啟以進行跨區域負載平衡。
6. 選擇儲存變更。

AWS CLI

開啟跨區域負載平衡

使用 modify-target-group-attributes 命令並將 `load_balancing.cross_zone.enabled` 屬性設為 `true`。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=load_balancing.cross_zone.enabled,Value=true"
```

CloudFormation

開啟跨區域負載平衡

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 `load_balancing.cross_zone.enabled` 屬性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.cross_zone.enabled"
          Value: "true"
```

自動目標權重 (ATW)

自動目標權重 (ATW) 會持續監控執行您應用程式的目標，偵測顯著的效能偏差，稱為異常。ATW 可透過即時資料異常偵測，動態調整路由至目標的流量。

自動目標權重 (ATW) 會自動對您帳戶中的每個 Application Load Balancer 執行異常偵測。識別異常目標時，ATW 可以透過減少路由的流量來自動嘗試穩定這些目標，稱為異常緩解。ATW 會持續最佳化流量分佈，以最大化每個目標的成功率，同時將目標群組失敗率降至最低。

考量：

- 異常偵測目前會監控來自目標的 HTTP 5xx 回應碼，以及目標的連線失敗。異常偵測一律開啟且無法關閉。
- 使用 Lambda 做為目標時，不支援 ATW。

內容

- 異常偵測
- 異常緩解

異常偵測

ATW 異常偵測會監控與其目標群組中其他目標的行為是否有顯著偏差的任何目標。這些偏差稱為異常，是透過比較一個目標的錯誤百分比與目標群組中其他目標的錯誤百分比來決定。這些錯誤可能是連線錯誤和 HTTP 錯誤代碼。然後，報告明顯高於其對等的目標會被視為異常。

異常偵測需要在目標群組中至少三個運作狀態良好的目標。當目標註冊到目標群組時，必須先通過運作狀態檢查，才能接收流量。目標開始接收流量後，ATW 會開始監控目標並持續發佈異常結果。對於沒有異常的目標，異常結果為 `normal`。對於具有異常的目標，異常結果為 `anomalous`。

ATW 異常偵測的運作與目標群組運作狀態檢查無關。目標可以通過所有目標群組運作狀態檢查，但由於錯誤率提高，仍然會標記為異常。變成異常的目標不會影響其目標群組運作狀態檢查狀態。

異常偵測狀態

您可以檢視目前的異常偵測狀態。以下為可能值：

- `normal` – 未偵測到異常。
- `anomalous` – 偵測到異常。

Console

檢視異常偵測狀態

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Targets (目標) 標籤。
5. 在已註冊目標資料表中，異常偵測結果欄會顯示每個目標的異常狀態。

AWS CLI

檢視異常偵測狀態

使用 describe-target-health 命令。下列範例顯示指定目標群組中每個目標的狀態。

```
aws elbv2 describe-target-health \
    --target-group-arn target-group-arn \
```

```
    --include AnomalyDetection
```

異常緩解

ATW 異常緩解會自動從異常目標路由流量，讓他們有機會復原。

需求

只有在使用加權隨機路由演算法時，才能使用 ATW 的異常緩解函數。

在緩解期間：

- ATW 會定期調整路由至異常目標的流量。目前，期間為每 5 秒一次。
- ATW 會將路由至異常目標的流量減少至執行異常緩解所需的最低數量。
- 不再偵測到為異常的目標會逐漸將更多流量路由到它們，直到它們與目標群組中的其他正常目標達到同位。

Console

開啟異常緩解

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 在流量組態窗格中，確認負載平衡演算法的選取值為隨機加權。

   一開始選取加權隨機演算法時，預設會開啟異常偵測。
6. 在異常緩解下，確保已選取開啟異常緩解。
7. 選擇儲存變更。

AWS CLI

開啟異常緩解

使用 modify-target-group-attributes 命令搭配 `load_balancing.algorithm.anomaly_mitigation` 屬性。

```
aws elbv2
```

緩解狀態

您可以檢查 ATW 是否正在對目標執行緩解措施。以下為可能值：

- yes – 緩解進行中。
- no – 緩解未進行中。

Console

檢視異常緩解狀態

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Targets (目標) 標籤。
5. 在已註冊目標資料表中，您可以在緩解生效欄中檢視每個目標的異常緩解狀態。

AWS CLI

檢視異常緩解狀態

使用 describe-target-health 命令。下列範例顯示指定目標群組中每個目標的狀態。

```
aws elbv2 describe-target-health \
    --target-group-arn target-group-arn \
    --include AnomalyDetection
```

黏性工作階段

根據預設，Application Load Balancer 會根據所選的負載平衡演算法，將每個請求獨立路由至註冊的目標。不過，您可以使用粘性會話功能 (也稱為工作階段親和性)，讓負載平衡器將使用者的工作階段繫結到特定目標。這樣能確保該工作階段期間所有的使用者請求都能傳送到同一個目標。這功能對於維護狀態資訊以便為用戶端提供持續體驗的伺服器來說很實用。若要使用粘性會話，用戶端必須支援 Cookie。

Application Load Balancer 支援持續時間型 Cookie 和應用程式型 Cookie。粘性會話會在目標群組層級啟用。您可以組合使用持續時間型粘性、應用程式型粘性，以及目標群體之間無粘性。

管理粘性會話的金鑰是決定您的負載平衡器應該持續將使用者請求路由到同一個目標的時間。如果您的應用程式有自己的工作階段 Cookie，則您可以使用應用程式型粘性，負載平衡器工作階段 Cookie 會遵循應用程式的工作階段 Cookie 指定的持續時間。如果您的應用程式沒有自己的工作階段 Cookie，則您可以使用持續時間型粘性，來產生具有指定持續時間的負載平衡器工作階段 Cookie。

系統會使用輪換金鑰來對負載平衡器產生的 Cookie 內容進行加密。您無法解密或修改負載平衡器產生的 Cookie。

對於這兩種粘性類型，Application Load Balancer 會在每次請求後重設其產生的 Cookie 到期時間。如果 Cookie 過期，則工作階段不再具有粘性，用戶端應該從其 Cookie 存放區中刪除該 Cookie。

要求

- HTTP/HTTPS 負載平衡器。
- 在各個可用區域內啟動至少一個正常運作的執行個體。

考量事項

- 如果跨區域負載平衡已停用，便不支援粘性會話。停用跨區域負載平衡時，嘗試啟用黏性工作階段失敗。
- 對於應用程式型 Cookie，必須針對每個目標群組個別指定 Cookie 名稱。但是，對於持續時間型 Cookie，AWSALB 是所有目標群組中唯一使用的名稱。
- 如果您使用多層 Application Load Balancer，則可以使用應用程式型 Cookie 在所有層級間啟用粘性會話。但是，如果使用持續時間型 Cookie，便只能在一個層上啟用粘性會話，因為 AWSALB 是唯一可用的名稱。
- 如果 Application Load Balancer 同時收到 AWSALBCORS和以AWSALB持續時間為基礎的黏性 Cookie，則 中的值AWSALBCORS將優先。
- 應用程式型粘性不適用於加權目標群組。
- 如果您有一個轉送規則涉及多個目標群組，且一個或多個目標群組已啟用粘性會話，則您必須啟用目標群組層級的粘性。
- WebSocket 連線本質上具粘性。如果用戶端請求將連線升級到 WebSocket，傳回 HTTP 101 狀態碼以接受連線升級的目標，為 WebSocket 連線中使用的目標。在 WebSocket 升級完成之後，不會使用以 Cookie 為基礎的黏性。
- Application Load Balancer 會使用 Cookie 標頭中的 Expires 屬性，而非 Max-Age 屬性。

- Application Load Balancer 不支援 URL 編碼的 Cookie 值。
- 如果 Application Load Balancer 在目標因取消註冊而耗盡時收到新的請求，則請求會路由至運作狀態良好的目標。
- 如果啟用目標最佳化工具，則不支援黏性工作階段。

黏性類型

- 持續時間型粘性
- 應用程式型粘性

持續時間型粘性

持續時間型粘性會使用負載平衡器產生的 Cookie (AWSALB)，將請求路由至目標群組中的同一個目標。Cookie 用於將工作階段映射到目標。如果您的應用程式沒有自己的工作階段 Cookie，您可以指定自己的粘性持續時間，並管理負載平衡器應持續地將使用者的請求路由至同一個目標的時間。

負載平衡器首次從用戶端收到請求時，它會將請求路由到目標 (根據所選演算法)，並產生一個名為 AWSALB 的 Cookie。它會對所選目標的資訊進行編碼，對 Cookie 進行加密，並在對用戶端的回應中包含 Cookie。負載平衡器產生的 Cookie 自己有 7 天的有效期，且無法設定。

在後續的請求中，用戶端應該包括 AWSALB Cookie。負載平衡器收到來自用戶端包含 Cookie 的請求時，會偵測到此 Cookie，並會將該請求路由至同一個目標。如果 Cookie 存在但無法解碼，或者如果它是指已取消註冊或運作狀態不佳的目標，負載平衡器會選取新目標，並使用新目標的相關資訊更新 Cookie。

對於跨來源資源共用 (CORS) 請求，某些瀏覽器需要 `SameSite=None; Secure`才能啟用黏性。為了支援這些瀏覽器，負載平衡器一律會產生第二個黏性 CookieAWSALBCORS，其中包含與原始黏性 Cookie 相同的資訊，以及 `SameSite` 屬性。用戶端會收到兩個 Cookie，包括非 CORS 請求。

Console

啟用以持續時間為基礎的黏性

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。

5. 在目標選取組態下，執行下列動作：

   a. 選取開啟黏性。

   b. 在粘性類型中，選取負載平衡器產生的 Cookie。

   c. 針對 Stickiness duration (黏性持續期間)，指定介於 1 秒到 7 天之間的值。

6. 選擇儲存變更。

AWS CLI

啟用以持續時間為基礎的黏性

使用 modify-target-group-attributes 命令搭配 `stickiness.enabled` 和 `stickiness.lb_cookie.duration_seconds` 屬性。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \
        "Key=stickiness.enabled,Value=true" \
        "Key=stickiness.lb_cookie.duration_seconds,Value=300"
```

CloudFormation

啟用以持續時間為基礎的黏性

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 `stickiness.enabled`和 `stickiness.lb_cookie.duration_seconds` 屬性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "stickiness.enabled"
          Value: "true"
        - Key: "stickiness.lb_cookie.duration_seconds"
```

```
        Value: "300"
```

應用程式型粘性

應用程式型粘性可讓您靈活地設定自己的用戶端目標粘性標準。啟用應用程式型粘性後，負載平衡器會根據選擇的演算法，將第一個請求路由到目標群組內的目標。目標預期會設定與負載平衡器上設定的 Cookie 相符的自訂應用程式 Cookie，以啟用粘性。這個自定義 Cookie 可以包括應用程序所需的任何 Cookie 屬性。

Application Load Balancer 從目標收到自訂應用程式 Cookie 後，會自動產生新的加密應用程式 Cookie，以擷取粘性資訊。此負載平衡器產生的應用程式 Cookie 會擷取每個已啟用應用程式型粘性之目標群組的粘性資訊。

負載平衡器產生的應用程式 Cookie 不會複製目標所設定之自訂 Cookie 的屬性。它自己有 7 天的有效期，且無法設定。在對用戶端的回應中，Application Load Balancer 只會驗證在目標群組層級設定之自訂 Cookie 的名稱，而不會驗證自訂 Cookie 的值或到期屬性。只要名稱相符，負載平衡器會在對用戶端待回應中傳送兩個 Cookie，即目標設定的自訂 Cookie 以及負載平衡器產生的應用程式 Cookie。

在後續請求中，用戶端必須傳回這兩個 Cookie 以保持粘性。負載平衡器會解密應用程式 Cookie，並檢查設定的粘性持續時間是否仍然有效。然後，它會使用 Cookie 中的資訊來將請求傳送給目標群組中的同一個目標，以維持粘性。負載平衡器也會將自訂應用程式 Cookie 代理至目標，且不會對其進行檢查或修改。在後續回應中，負載平衡器產生的應用程式 Cookie 到期時間，以及在負載平衡器上設定的粘性持續時間都會重設。為了保持用戶端和目標之間的粘性，Cookie 的到期時間和粘性的持續時間不應結束。

如果目標失敗或運作狀態不佳，負載平衡器會停止路由請求到該目標，並根據選擇的負載平衡演算法選擇運作狀態良好的新目標。負載平衡器會將工作階段視為「粘到」運作狀態良好的新目標，並持續路由請求到運作狀態良好的新目標，即使失敗的目標恢復也是如此。

對於跨來源資源共用 (CORS) 請求，若要啟用粘性，負載平衡器只會在使用者代理程式版本為 Chromium80 或更高版本時，將 `SameSite=None; Secure` 屬性新增至負載平衡器產生的應用程式 Cookie。

由於大多數瀏覽器將 Cookie 的大小限制為 4K，負載平衡器會將大於 4K 的應用程式 Cookie 分割成多個 Cookie。Application Load Balancer 支援的 Cookie 大小上限為 16K，因此最多會建立 4 個傳送到用戶端的碎片。用戶端看到的應用程式 Cookie 名稱以 "AWSALBAPP-" 開頭，並包含片段編號。例如，如果 Cookie 大小為 0-4K，則用戶端會看到 AWSALBAPP-0。如果 Cookie 大小是 4-8K，則用戶端會看到 AWSALBAPP-0 和 AWSALBAPP-1，依此類推。

Console

啟用應用程式型黏性

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 在目標選取組態下，執行下列動作：
   a. 選取開啟黏性。
   b. 在粘性類型中，選取應用程式型 Cookie。
   c. 針對 Stickiness duration (黏性持續期間)，指定介於 1 秒到 7 天之間的值。
   d. 在應用程式 Cookie 名稱中，輸入應用程式型 Cookie 的名稱。

      請勿使用 AWSALB、AWSALBAPP、或 AWSALBTG 作為 Cookie 名稱；它們預留供負載平衡器使用。
6. 選擇儲存變更。

AWS CLI

啟用應用程式型黏性

使用 modify-target-group-attributes 命令搭配以下屬性：

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \
        "Key=stickiness.enabled,Value=true" \
        "Key=stickiness.type,Value=app_cookie" \
        "Key=stickiness.app_cookie.cookie_name,Value=my-cookie-name" \
```

```
        "Key=stickiness.app_cookie.duration_seconds,Value=300"
```

CloudFormation

啟用應用程式型黏性

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含下列屬性：

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "stickiness.enabled"
          Value: "true"
        - Key: "stickiness.type"
          Value: "app_cookie"
        - Key: "stickiness.app_cookie.cookie_name"
          Value: "my-cookie-name"
        - Key: "stickiness.app_cookie.duration_seconds"
          Value: "300"
```

手動重新平衡

縱向擴展時，如果目標數量大幅增加，則由於粘性，可能會導致負載分配不均衡。在這種情況下，可以使用下列兩個選項重新平衡目標上的負載：

- 在由應用程式產生的 Cookie 上設定早於目前日期和時間的到期時間。這可防止用戶端將 Cookie 傳送至 Application Load Balancer，這會重新啟動建立黏性的程序。

- 在負載平衡器的應用程式型黏性組態上設定較短的持續時間，例如 1 秒。即使目標設定的 Cookie 未過期，這也會強制 Application Load Balancer 重新建立黏性。

向 Application Load Balancer 目標群組註冊目標

您會向目標群組註冊您的目標。建立目標群組時，您會指定其目標類型，這會決定您目標的註冊方式。例如，您可以註冊執行個體 ID、IP 地址或 Lambda 函數。如需詳細資訊，請參閱Application Load Balancer 的目標群組。

如果對目前已註冊目標的需求增加，您可以註冊額外的目標來應付需求。當目標準備好處理請求時，請透過目標群組來註冊目標。在註冊程序完成、目標通過初始的運作狀態檢查之後，負載平衡器就會立即開始將請求轉送到目標。

如果對已註冊目標的需求減少，或是需要為目標提供服務，您可以從目標群組取消目標的註冊。取消目標的註冊之後，負載平衡器就會立即停止將請求轉送到目標。當目標準備好接收請求時，您可以再次將目標註冊到目標群組。

當您取消註冊目標時，負載平衡器會等到傳輸中的請求完成。這稱為連接耗盡。當連接耗盡作業正在進行時，目標的狀態是 `draining`。

取消註冊透過 IP 地址註冊的目標時，您必須等待取消註冊延遲完成，之後才能再次註冊相同的 IP 地址。

如果是根據執行個體 ID 來註冊目標，您可以使用負載平衡器搭配 Auto Scaling 群組。在將目標群組連接到 Auto Scaling 群組，而且群組擴展之後，由 Auto Scaling 群組啟動的執行個體會自動註冊到目標群組。如果分離目標群組與 Auto Scaling 群組的連結，會自動從該目標群組中取消註冊執行個體。如需詳細資訊，請參閱 Amazon EC2 Auto Scaling User Guide 中的 Attaching a load balancer to your Auto Scaling group。

關閉目標上的應用程式時，您必須先從目標群組取消註冊目標，並預留時間讓現有連線耗盡。您可以使用 describe-target-health CLI 命令或重新整理 中的目標群組檢視來監控取消註冊狀態 AWS 管理主控台。確認目標取消註冊後，您可以繼續停止或終止應用程式。此序列可防止使用者在處理流量時，於應用程式終止時遇到 5XX 錯誤。

目標安全群組

當您將 EC2 執行個體註冊為目標時，必須確定執行個體的安全群組，會允許負載平衡器同時透過接聽程式連接埠和運作狀態檢查連接埠與您的執行個體通訊。

建議的規則

Inbound

來源	連接埠範圍	Comment
#########	########	Allow traffic from the load balancer on the instance listener port
#########	######	Allow traffic from the load balancer on the health check port

我們也建議您允許傳入 ICMP 流量，以支援路徑 MTU 探索。如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的路徑 MTU 探索。

目標最佳化工具

目標最佳化工具可讓您對目標群組中的目標強制執行嚴格的並行。它使用您在目標上安裝和設定的代理程式的協助。代理程式可做為負載平衡器和應用程式之間的內嵌代理。您可以將代理程式設定為強制執行負載平衡器可傳送至目標的並行請求數量上限。代理程式會追蹤目標正在處理的請求數量。當數字低於設定的最大值時，代理程式會傳送訊號給負載平衡器，讓它知道目標已準備好處理另一個請求。

若要啟用目標最佳化工具，您可以在建立目標群組時指定目標控制連接埠。負載平衡器會與此連接埠上的代理程式建立控制頻道，以管理流量。此連接埠與負載平衡器傳送應用程式流量的連接埠不同。向目標群組註冊的目標必須讓代理程式在其上執行。

注意：目標最佳化工具只能在建立目標群組期間啟用。目標控制連接埠無法在建立後修改。

代理程式以 Docker 影像的形式提供於： `public.ecr.aws/aws-elb/target-optimizer/target-control-agent:latest`。您可以在執行代理程式容器時設定下列環境變數：

`TARGET_CONTROL_DATA_ADDRESS`

代理程式會從此通訊端 (IP：port) 上的負載平衡器接收應用程式流量。此通訊端中的連接埠是您為目標群組設定的應用程式流量連接埠。根據預設，代理程式可以同時接受純文字和 TLS 連線。

TARGET_CONTROL_CONTROL_ADDRESS

代理程式會從此通訊端 (IP：port) 上的負載平衡器接收管理流量。通訊端中的連接埠是您為目標群組設定的目標控制連接埠。

TARGET_CONTROL_DESTINATION_ADDRESS

代理程式會將應用程式流量代理到此通訊端 (IP：port)。您的應用程式應該正在接聽此通訊端。

(選用) TARGET_CONTROL_MAX_CONCURRENCY

目標將從負載平衡器接收的並行請求數目上限。它可以介於 0-1000 之間。預設為 1。

(選用) TARGET_CONTROL_TLS_CERT_PATH

代理程式在 TLS 交握期間提供給負載平衡器的 TLS 憑證位置。根據預設，代理程式會在記憶體中產生自我簽署的憑證。

(選用) TARGET_CONTROL_TLS_KEY_PATH

對應至代理程式在 TLS 交握期間提供給負載平衡器之 TLS 憑證的私有金鑰位置。根據預設，代理程式會在記憶體中產生私有金鑰。

(選用) TARGET_CONTROL_TLS_SECURITY_POLICY

您為目標群組設定的 ELB 安全政策。預設值為 ELBSecurityPolicy-2016-08。

(選用) TARGET_CONTROL_PROTOCOL_VERSION

負載平衡器與代理程式通訊的通訊協定。可能的值為 HTTP1、 HTTP2、 GRPC。預設值為 HTTP1。

(選用) RUST_LOG

代理程式程序的日誌層級。代理程式軟體是以 Rust 撰寫。可能的值為 debug、 info和 error 。預設值為 info。

若要修改任何環境變數的值，您必須使用新值重新啟動代理程式。
您可以使用下列指標監控目標最佳化工具：TargetControlRequestCount、TargetControlRequestRejectCount、TargetControlActiv TargetControlWorkQueueLength、、TargetControlProcessedBytes。如需詳細資訊，請參閱目標最佳化工具指標 如需故障診斷資訊，請參閱目標最佳化工具故障診斷

共用子網路

參與者可以在共用 VPC 中建立 Application Load Balancer。參與者無法註冊在未與他們共用的子網路中執行的目標。

登記目標

在負載平衡器能夠使用的每個可用區域中，每個目標群組都必須擁有至少一個已登錄的目標。

目標群組的目標類型會決定您向該目標群組註冊目標的方式。如需詳細資訊，請參閱Target type (目標類型)。

需求和考量事項

- 在註冊時，執行個體必須處於 `running` 狀態。
- 目標執行個體必須位於您為目標群組指定的虛擬私有雲端 (VPC) 中。
- 依執行個體 ID 註冊 IPv6 目標群組的目標時，目標必須具有指派的主要 IPv6 地址。若要進一步了解，請參閱《Amazon EC2 使用者指南》中的 IPv6 地址
- 依 IPv4 目標群組的 IP 地址註冊目標時，您註冊的 IP 地址必須來自下列其中一個 CIDR 區塊：
  - 目標群組 VPC 的子網路
  - 10.0.0.0/8 (RFC 1918)
  - 100.64.0.0/10 (RFC 6598)
  - 172.16.0.0/12 (RFC 1918)
  - 192.168.0.0/16 (RFC 1918)
- 依 IPv6 目標群組的 IP 地址註冊目標時，您註冊的 IP 地址必須位於 VPC IPv6 CIDR 區塊內或對等 VPC 的 IPv6 CIDR 區塊內。
- 您無法在相同的 VPC 中註冊另一個 Application Load Balancer 的 IP 地址。如果另一個 Application Load Balancer 位於與負載平衡器 VPC 對等的 VPC 中，您可以註冊其 IP 地址。

Console

註冊目標

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。

3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Targets (目標) 標籤。
5. 選擇 Register targets (註冊目標)。
6. 如果目標群組的目標類型為 `instance`，請選取可用的執行個體，視需要覆寫預設連接埠，然後選擇包含為待定。
7. 如果目標群組的目標類型為 `ip`，請針對每個 IP 地址選取網路，輸入 IP 地址和連接埠，然後選擇包含為以下待定。
8. 如果目標群組的目標類型是 `lambda`，請選取 Lambda 函數或輸入其 ARN。如需詳細資訊，請參閱使用 Lambda 函數做為目標。
9. 選擇註冊待定目標。

AWS CLI

註冊目標

使用 register-targets 命令。下列範例會依執行個體 ID 註冊目標。由於未指定連接埠，負載平衡器會使用目標群組連接埠。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

下列範例依 IP 地址註冊目標。由於未指定連接埠，負載平衡器會使用目標群組連接埠。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=10.0.50.10 Id=10.0.50.20
```

下列範例會將 Lambda 函數註冊為目標。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

CloudFormation

註冊目標

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含新目標。下列範例會依執行個體 ID 註冊兩個目標。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
        - Id: !GetAtt Instance1.InstanceId
          Port: 80
        - Id: !GetAtt Instance2.InstanceId
          Port: 80
```

取消註冊目標

如果對您應用程式的需求減少，或者您需要為目標提供服務，可以從目標群組取消目標的登錄。取消目標的登錄，會將該目標從目標群組中移除，但不會影響到目標。

Console

取消註冊目標

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在目標索引標籤上，選取要移除的目標。
5. 選擇 Deregister (取消註冊)。
6. 出現確認的提示時，請選擇取消註冊。

AWS CLI

取消註冊目標

使用 deregister-targets 命令。下列範例會取消註冊透過執行個體 ID 註冊的兩個目標。

```
aws elbv2 deregister-targets \
    --target-group-arn target-group-arn \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

使用 Lambda 函數作為 Application Load Balancer 的目標

您可以將 Lambda 函數註冊為目標，並設定接聽程式規則，將請求轉送到 Lambda 函數的目標群組。當負載平衡器將請求轉送到使用 Lambda 函數做為目標的目標群組時，它會呼叫您的 Lambda 函數，並將請求的內容以 JSON 格式傳遞至 Lambda 函數。

負載平衡器會直接叫用 Lambda 函數，而不是使用網路連線。因此，Application Load Balancer 安全群組的傳出規則沒有要求。

限制

- Lambda 函數和目標群組必須在相同的帳戶中，且在相同的區域內。
- 您可以傳送到 Lambda 函數之請求內文的大小上限是 1 MB。如需相關的大小限制，請參閱 HTTP header limits。
- Lambda 函數可以傳送的回應 JSON 的大小上限是 1 MB。
- 不支援 WebSocket。升級請求會被拒絕，出現 HTTP 400 代碼。
- 不支援 Local Zone。
- 不支援自動目標權重 (ATW)。

目錄

- 準備 Lambda 函數
- 為 Lambda 函數建立目標群組
- 從負載平衡器接收事件
- 對負載平衡器進行回應
- 多值標頭
- 啟用運作狀態檢查
- 註冊 Lambda 函數
- 取消註冊 Lambda 函數

如需示範，請參閱 Lambda Target on Application Load Balancer。

準備 Lambda 函數

如果將 Lambda 函數與 Application Load Balancer 搭配使用，請採用下列建議。

調用 Lambda 函數的許可

如果您使用 AWS 管理主控台來建立目標群組和註冊 Lambda 函數，主控台會代表您將所需許可新增至 Lambda 函數政策。否則，在建立目標群組並使用 註冊函數之後 AWS CLI，您必須使用 add-permission 命令授予 Elastic Load Balancing 調用 Lambda 函數的許可。我們建議您使用 `aws:SourceAccount` 和 `aws:SourceArn` 條件索引鍵來將函數調用限制在指定的目標群組。如需詳細資訊，請參閱《IAM 使用者指南》中的混淆代理人問題。

```
aws lambda add-permission \
    --function-name lambda-function-arn-with-alias-name \
    --statement-id elb1 \
    --principal elasticloadbalancing.amazonaws.com \
    --action lambda:InvokeFunction \
    --source-arn target-group-arn \
    --source-account target-group-account-id
```

Lambda 函數版本控制

您可以為每個目標群組註冊一個 Lambda 函數。為了確保您可以變更 Lambda 函數，且負載平衡器一律會呼叫目前版本的 Lambda 函數，請建立一個函數別名，並將該別名包含在向負載平衡器註冊 Lambda 函數時的函數 ARN 中。如需詳細資訊，請參閱《 AWS Lambda 開發人員指南》中的AWS Lambda 函數別名。

函數逾時

負載平衡器會等待直到您的 Lambda 函數回應或逾時。建議您根據您預期的執行時間來設定 Lambda 函數逾時。如需預設逾時值及其變更方式的相關資訊，請參閱設定 Lambda 函數逾時。如需您可以設定的逾時值上限的相關資訊，請參閱AWS Lambda 配額。

為 Lambda 函數建立目標群組

建立目標群組以用於請求路由。如果請求內容符合某接聽程式規則，內含會將它轉送到此目標群組的動作，則負載平衡器會呼叫註冊的 Lambda 函數。

Console

建立目標群組和註冊 Lambda 函數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇 Create target group (建立目標群組)。
4. 在選取目標類型中，選取 Lambda 函數。
5. 針對 Target group name (目標群組名稱)，輸入目標群組的名稱。
6. (選用) 若要啟用運作狀態檢查，請選擇運作狀態檢查區段中的啟用。
7. （選用）展開標籤。針對每個標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
8. 選擇下一步。
9. 如果您準備好註冊 Lambda 函數，請選擇選取 Lambda 函數，然後從清單中選擇 Lambda 函數，或選擇輸入 Lambda 函數 ARN，然後輸入 Lambda 函數的 ARN。

   如果您尚未準備好註冊 Lambda 函數，請稍後選擇註冊 Lambda 函數，稍後再註冊目標。如需詳細資訊，請參閱the section called “登記目標”。
10. 選擇 Create target group (建立目標群組)。

AWS CLI

建立 類型的目標群組 lambda

使用 create-target-group 命令。

```
aws elbv2 create-target-group \
    --name my-target-group \
    --target-type lambda
```

註冊 Lambda 函數

使用 register-targets 命令。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

CloudFormation

建立目標群組和註冊 Lambda 函數

定義 AWS::ElasticLoadBalancingV2::TargetGroup 類型的資源。如果您現在尚未準備好註冊 Lambda 函數，您可以省略 Targets 屬性，稍後再將其新增。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
```

從負載平衡器接收事件

負載平衡器同時支援透過 HTTP 和 HTTPS 的請求進行 Lambda 呼叫。負載平衡器會以 JSON 格式傳送事件。負載平衡器會將以下標頭新增至每個請求：X-Amzn-Trace-Id、X-Forwarded-For、X-Forwarded-Port 和 X-Forwarded-Proto。

如果 content-encoding 標頭存在，負載平衡器 Base64 會對內文使用 Base64 編碼並將 isBase64Encoded 設定為 true。

如果 content-encoding 標頭不存在，則 Base64 編碼取決於內容類型。如果是以下類型，負載平衡器會依原樣傳送內文並將 isBase64Encoded 設定為 false：text/*、application/json、application/javascript 和 application/xml。否則，負載平衡器會對內文使用 Base64 編碼，並將 isBase64Encoded 設定為 true。

以下為範例 事件。

```
{
    "requestContext": {
        "elb": {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
group/6d0ecf831eec9f09"
```

```
        }
    },
    "httpMethod": "GET",
    "path": "/",
    "queryStringParameters": {parameters},
    "headers": {
        "accept": "text/html,application/xhtml+xml",
        "accept-language": "en-US,en;q=0.8",
        "content-type": "text/plain",
        "cookie": "cookies",
        "host": "lambda-846800462-us-east-2.elb.amazonaws.com",
        "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)",
        "x-amzn-trace-id": "Root=1-5bdb40ca-556d8b0c50dc66f0511bf520",
        "x-forwarded-for": "72.21.198.66",
        "x-forwarded-port": "443",
        "x-forwarded-proto": "https"
    },
    "isBase64Encoded": false,
    "body": "request_body"
}
```

對負載平衡器進行回應

來自 Lambda 函數的回應必須包含 Base64 編碼狀態、狀態碼、狀態描述和標頭。您可以省略內文。

若要在回應的內文中包含二進位內容，您必須將內容以 Base64 編碼，並將 `isBase64Encoded` 設定為 `true`。負載平衡器會解碼內容，以擷取二進位內容，並將其傳送至 HTTP 回應內文中的用戶端。

負載平衡器並不遵守按躍點的標頭，例如 `Connection` 或 `Transfer-Encoding`。您可以省略 `Content-Length` 標頭，因為負載平衡器會在將回應傳送至用戶端之前計算。

以下是來自基於 Lambda 函數的 nodejs 的範例回應。

```
{
    "isBase64Encoded": false,
    "statusCode": 200,
    "statusDescription": "200 OK",
    "headers": {
        "Set-cookie": "cookies",
        "Content-Type": "application/json"
    },
    "body": "Hello from Lambda (optional)"
```

```
}
```

若是與搭配 Application Load Balancer 使用的 Lambda 函數範本，相關資訊請至 GitHub 參閱 application-load-balancer-serverless-app。或者，開啟 Lambda 主控台，選擇應用程式、建立應用程式，然後從 AWS Serverless Application Repository中選取下列其中一項：

- ALB-Lambda-Target-UploadFiletoS3
- ALB-Lambda-Target-BinaryResponse
- ALB-Lambda-Target-WhatisMyIP

多值標頭

如果來自用戶端的請求或來自 Lambda 函數的回應，包含具有多個值的標頭或包含相同標頭多次，或查詢參數具有多個值的相同索引鍵，您可以啟用對多值標頭語法的支援。啟用多值標頭之後，負載平衡器和 Lambda 函數之間交換的標頭和查詢參數會使用陣列而不是字串。如果您未啟用多值標頭語法，且標頭或查詢參數具有多個值，負載平衡器會使用它接收的最後一個值。

目錄

- 具有多值標頭的請求
- 具有多值標頭的回應
- 啟用多值標頭

具有多值標頭的請求

根據您是否為目標群組啟用多值標頭而定，用於標頭和查詢字串參數的欄位名稱有所不同。

以下範例請求具有使用相同金鑰的兩個查詢參數：

```
http://www.example.com?&myKey=val1&myKey=val2
```

採用預設格式時，負載平衡器會使用用戶端傳送的最後一個值，並使用 `queryStringParameters` 向您傳送包含查詢字串參數的事件。例如：

```
"queryStringParameters": { "myKey": "val2"},
```

如果您啟用多值標頭，負載平衡器會使用用戶端傳送的兩個金鑰值，並使用 `multiValueQueryStringParameters` 向您傳送包含查詢字串參數的事件。例如：

```
"multiValueQueryStringParameters": { "myKey": ["val1", "val2"] },
```

同樣地，假設用戶端會傳送的請求標頭中具有兩個 Cookie：

```
"cookie": "name1=value1",
"cookie": "name2=value2",
```

採用預設格式時，負載平衡器會使用用戶端傳送的最後一個 Cookie，並使用 `headers` 向您傳送包含標頭的事件。例如：

```
"headers": {
    "cookie": "name2=value2",
    ...
},
```

如果您啟用多值標頭，負載平衡器會使用用戶端傳送的兩個 Cookie，並使用 `multiValueHeaders` 向您傳送包含標頭的事件。例如：

```
"multiValueHeaders": {
    "cookie": ["name1=value1", "name2=value2"],
    ...
},
```

如果查詢參數是 URL 編碼，負載平衡器不會進行解碼。您必須在 Lambda 函數中解碼。

具有多值標頭的回應

根據您是否為目標群組啟用多值標頭而定，用於標頭的欄位名稱有所不同。如果您已啟用多重值標頭和 `headers`，您必須使用 `multiValueHeaders`。

使用預設格式，您可以指定單一 Cookie：

```
{
  "headers": {
      "Set-cookie": "cookie-name=cookie-value;Domain=myweb.com;Secure;HttpOnly",
      "Content-Type": "application/json"
  },
}
```

如果您啟用多值標頭，您必須如下所示指定多個 Cookie：

```
{
  "multiValueHeaders": {
      "Set-cookie": ["cookie-name=cookie-
value;Domain=myweb.com;Secure;HttpOnly","cookie-name=cookie-value;Expires=May 8,
 2019"],
      "Content-Type": ["application/json"]
  },
}
```

負載平衡器可能會依照與 Lambda 回應承載中指定順序不同的順序將標頭傳送到用戶端。因此，不要指望標頭會以特定順序返回。

啟用多值標頭

您可以為具有目標類型 `lambda` 的目標群組啟用或停用多值標頭。

Console

啟用多值標頭

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 啟用多值標頭。
6. 選擇儲存變更。

AWS CLI

啟用多值標頭

使用 modify-target-group-attributes 命令搭配 `lambda.multi_value_headers.enabled` 屬性。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=lambda.multi_value_headers.enabled,Value=true"
```

CloudFormation

啟用多值標頭

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 `lambda.multi_value_headers.enabled` 屬性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
      TargetGroupAttributes:
        - Key: "lambda.multi_value_headers.enabled"
          Value: "true"
```

啟用運作狀態檢查

在預設情況下，會為類型 `lambda` 的目標群組停用運作狀態檢查。您可以啟用運作狀態檢查，以便使用 Amazon Route 53 來實作 DNS 備援。Lambda 函數可以檢查下游服務的運作狀態，之後再回應運作狀態檢查的請求。如果來自 Lambda 函數的回應指出運作狀態檢查失敗，則會將運作狀態檢查失敗傳遞至 Route 53。您可以設定 Route 53 以容錯移轉到備用的應用程式堆疊。

將向您就任何 Lambda 函數呼叫而進行的運作狀態檢查收費。

以下是傳送到 Lambda 函數的運作狀態檢查事件格式。若要檢查事件是否為運作狀態檢查事件，請檢查 user-agent 欄位的值。運作狀態檢查的使用者代理程式為 `ELB-HealthChecker/2.0`。

```
{
    "requestContext": {
        "elb": {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
group/6d0ecf831eec9f09"
```

```
        }
    },
    "httpMethod": "GET",
    "path": "/",
    "queryStringParameters": {},
    "headers": {
        "user-agent": "ELB-HealthChecker/2.0"
    },
    "body": "",
    "isBase64Encoded": false
}
```

Console

啟用lambda目標群組的運作狀態檢查

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在 Health checks (運作狀態檢查) 標籤上，選擇 Edit (編輯)。
5. 在運作狀態檢查中，選取啟用。
6. （選用）視需要更新運作狀態檢查設定。
7. 選擇儲存變更。

AWS CLI

啟用lambda目標群組的運作狀態檢查

使用 modify-target-group 命令。

```
aws elbv2 modify-target-group \
    --target-group-arn target-group-arn \
    --health-check-enabled
```

CloudFormation

啟用lambda目標群組的運作狀態檢查

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      HealthCheckEnabled: true
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
```

註冊 Lambda 函數

您可以在每個目標群組中註冊單一 Lambda 函數。若要取代 Lambda 函數，建議您建立新的目標群組、向新目標群組註冊新函數，以及更新接聽程式規則以使用新的目標群組。

Console

取消註冊 Lambda 函數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在目標索引標籤上，如果沒有註冊 Lambda 函數，請選擇註冊目標。
5. 選取 Lambda 函數或輸入其 ARN。
6. 選擇註冊。

AWS CLI

取消註冊 Lambda 函數

使用 register-targets 命令。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

CloudFormation

取消註冊 Lambda 函數

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      TargetType: lambda
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myLambdaFunction
```

取消註冊 Lambda 函數

如果您不再需要將流量傳送到您的 Lambda 函數，則可以將它取消註冊。取消註冊 Lambda 函數之後，傳輸中的請求會失敗，出現 HTTP 5XX 錯誤。

若要取代 Lambda 函數，建議您建立新的目標群組、向新目標群組註冊新函數，以及更新接聽程式規則以使用新的目標群組。

Console

取消註冊 Lambda 函數

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在目標索引標籤上，選取目標，然後選擇取消註冊。
5. 出現確認的提示時，請選擇取消註冊。

AWS CLI

取消註冊 Lambda 函數

使用 deregister-targets 命令。

```
aws elbv2 deregister-targets \
    --target-group-arn target-group-arn \
    --targets Id=lambda-function-arn
```

Application Load Balancer 目標群組的標籤

標籤可幫助您以不同的方式來將目標群組分類，例如，根據目的、擁有者或環境。

您可以在每個目標群組中加入多個標籤。每個目標群組的標籤索引鍵必須是唯一的。如果所新增的標籤，其索引鍵已經和目標群組具有關聯，則此動作會更新該標籤的值。

當您使用完標籤之後，可以將其移除。

限制

- 每一資源標籤數上限：50
- 索引鍵長度上限：127 個 Unicode 字元
- 數值長度上限：255 個 Unicode 字元
- 標籤鍵與值皆區分大小寫。允許的字元包括可用 UTF-8 表示的英文字母、空格和數字，還有以下特殊字元：+ - = . _ : / @。不可使用結尾或前方空格。
- 請勿在標籤名稱或值中使用 aws: 字首，因為它保留供 AWS 使用。您不可編輯或刪除具此字首的標籤名稱或值。具此字首的標籤，不算在受資源限制的標籤計數內。

Console

管理目標群組的標籤

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的 Load Balancing (負載平衡) 中，選擇 Target Groups (目標群組)。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在標籤索引標籤上，選擇管理標籤，並執行下列一個或多個動作：
   a. 若要更新標籤，請為索引鍵和值輸入新值。
   b. 如要新增標籤，請選擇新增標籤，然後輸入索引鍵和值的值。

c. 若要移除標籤，請選擇標籤旁的移除。

5. 選擇儲存變更。

AWS CLI

新增 標籤

使用 add-tags 命令。下列範例會新增兩個標籤。

```
aws elbv2 add-tags \
    --resource-arns target-group-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

移除標籤

使用 remove-tags 命令。下列範例會移除具有指定金鑰的標籤。

```
aws elbv2 remove-tags \
    --resource-arns target-group-arn \
    --tag-keys project department
```

CloudFormation

新增 標籤

更新 AWS::ElasticLoadBalancingV2::TargetGroup 資源以包含 Tags 屬性。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      Tags:
        - Key: 'project'
          Value: 'lima'
        - Key: 'department'
          Value: 'digital-media'
```

刪除 Application Load Balancer 目標群組

如果沒有任何接聽程式規則的轉送動作參照某目標群組，即可刪除該目標群組。刪除目標群組不會影響透過該目標群組登錄的目標。如果不再需要註冊的 EC2 執行個體，則可以停止或終止它。

Console

刪除目標群組

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格的負載平衡中，選擇目標群組。
3. 選取目標群組，然後依序選擇 Actions (動作)、Delete (刪除)。
4. 選擇 刪除。

AWS CLI

刪除目標群組

使用 delete-target-group 指令。

```
aws elbv2 delete-target-group \
    --target-group-arn target-group-arn
```

監控 Application Load Balancer

您可使用以下功能來監控負載平衡器、分析流量模式並對與負載平衡器和目標相關的問題進行疑難排解。

CloudWatch 指標

您可以使用 Amazon CloudWatch 來為負載平衡器擷取關於資料點的統計資料，並以一組按順序排列的時間序列資料為目標，也就是指標。您可以使用這些指標來確認您的系統是否依照預期執行。如需詳細資訊，請參閱適用於 Application Load Balancer 的 CloudWatch 指標。

存取日誌

您可以使用存取日誌，來擷取對負載平衡器發出之請求的詳細資訊，並將這些資訊作為日誌檔案存放在 Amazon S3。您可以使用這些存取日誌來分析流量模式，並排除目標的問題。如需詳細資訊，請參閱Application Load Balancer 的存取日誌。

連線日誌

您可以使用連線日誌來擷取傳送至負載平衡器之請求的屬性，並將其儲存為 Amazon S3 中的日誌檔案。您可以使用這些連線日誌來判斷使用的用戶端 IP 地址和連接埠、用戶端憑證資訊、連線結果和 TLS 密碼。然後，這些連線日誌可用於檢閱請求模式和其他趨勢。如需詳細資訊，請參閱Application Load Balancer 的連線日誌。

運作狀態檢查日誌

您可以使用運作狀態檢查日誌來擷取對負載平衡器註冊目標所做的運作狀態檢查詳細資訊，並將其儲存為 Amazon S3 中的日誌檔案。您可以使用這些運作狀態檢查日誌來疑難排解目標的問題。如需詳細資訊，請參閱運作狀態檢查日誌。

請求追蹤

您可以使用請求追蹤來追蹤 HTTP 請求。負載平衡器會在收到的每個請求中新增標頭和追蹤識別符。如需詳細資訊，請參閱Application Load Balancer 上的請求追蹤。

CloudTrail 日誌

您可以使用 AWS CloudTrail 擷取對 Elastic Load Balancing API 進行呼叫的詳細資訊，並將其儲存為 Amazon S3 中的日誌檔案。您可以使用這些 CloudTrail 日誌來判斷提出了哪些呼叫、提出呼叫的來源 IP 地址、提出呼叫的人員及時間等。如需詳細資訊，請參閱使用 CloudTrail 記錄 Elastic Load Balancing 的 API 呼叫。

適用於 Application Load Balancer 的 CloudWatch 指標

Elastic Load Balancing 會將負載平衡器和目標的資料點發佈到 Amazon CloudWatch。CloudWatch 可讓使用一組時間序列資料的形式來擷取這些資料點的相關統計資料，也就是指標。您可以將指標視為要監控的變數，且資料點是該變數在不同時間點的值。例如，您可以監控負載平衡器在一段指定期間內的運作狀態良好的目標總數量。每個資料點都有關聯的時間戳記和可選的測量單位。

您可以使用指標來確認系統的運作符合預期。例如，若指標超過您認為能夠接受的範圍，您可以建立 CloudWatch 警示來監控指定的指標並執行動作 (例如傳送通知到電子郵件地址)。

Elastic Load Balancing 只會在請求穿越負載平衡器時回報指標到 CloudWatch。如果有請求進出負載平衡器，Elastic Load Balancing 會以 60 秒為間隔來測量並傳送其指標。如果沒有請求流經負載平衡器，或者指標沒有資料，則不會回報該指標。

Application Load Balancer 的指標會排除運作狀態檢查請求。

如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

目錄

- Application Load Balancer 指標
- Application Load Balancer 的指標維度
- Application Load Balancer 指標的統計資料
- 檢視負載平衡器的 CloudWatch 指標

Application Load Balancer 指標

- 負載平衡器
- LCUs
- 目標
- 目標群組運作狀態
- Lambda 函數
- 使用者身分驗證
- 目標最佳化工具

AWS/ApplicationELB 命名空間包含下列負載平衡器指標。

指標	Description
ActiveConnectionCount	從用戶端到負載平衡器以及從負載平衡器到目標的並行作用中 TCP 連線總數。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
BYoIPUtilPercentage	IP 集區的用量百分比。 報告條件：負載平衡器上已啟用 BYoIP。 統計資訊：唯一有意義的統計資訊是 Average。 維度 • LoadBalancer ，TargetGroup • LoadBalancer ，TargetGroup ，AvailabilityZone
ClientTLSNegotiationErrorCount	由於 TLS 錯誤而未與負載平衡器建立工作階段之用戶端所啟動的 TLS 連線數目。可能的原因包括密碼或通訊協定不相符，或用戶端無法驗證伺服器憑證並關閉連線。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
DesyncMitigationMode_NonCom	不符合 RFC 7230 的請求數量。 報告條件：有非零值

指標	Description
pliant_Request_Count	統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
DroppedInvalidHeaderRequestCount	在傳送請求之前，負載平衡器移除具有無效標頭欄位的 HTTP 標頭的請求數目。只有在 routing.http.drop_invalid_header_fields.enabled 屬性設定為 true 時，負載平衡器才會移除這些標頭。 報告條件：有非零值 統計資訊：全部 維度 • AvailabilityZone ，LoadBalancer
ForwardedInvalidHeaderRequestCount	由 HTTP 標頭具有無效標頭欄位的負載平衡器所傳送的請求數目。只有在 routing.http.drop_invalid_header_fields.enabled 屬性設定為 false 時，負載平衡器才會轉送具有這些標頭的請求。 報告條件：一律報告 統計資訊：全部 維度 • AvailabilityZone ，LoadBalancer

指標	Description
GrpcRequestCount	透過 IPv4 與 IPv6 處理的 gRPC 請求數量。 報告條件：有非零值 統計資料：最實用的統計數量是 Sum。Minimum、Maximum 和 Average 都會傳回 1。 維度 • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup • TargetGroup • AvailabilityZone ，TargetGroup
HTTP_Fixed_Response_Count	成功的固定回應動作次數。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTP_Redirect_Count	成功的重新導向動作次數。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
HTTP_Redirect_Url_Limit_Exceeded_Count	因為回應位置標頭中的 URL 大於 8K 而無法完成的重新導向動作數。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_3XX_Count	源自於負載平衡器的 HTTP 3XX 重新導向代碼數目。此計數未包含目標所產生的回應碼。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
HTTPCode_ELB_4XX_Count	源自於負載平衡器的 HTTP 4XX 用戶端錯誤碼數目。此計數未包含目標所產生的回應碼。 要求的格式不正確或不完整時，會產生用戶端錯誤。除了負載平衡器傳回 HTTP 460 錯誤碼的情況之外，目標沒有收到這些要求。此計數未包含目標所產生的任何回應碼。 報告條件：有非零值 統計資料：最實用的統計數量是 Sum。Minimum、Maximum 和 Average 都會傳回 1。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_5XX_Count	源自於負載平衡器的 HTTP 5XX 伺服器錯誤碼數目。此計數未包含目標所產生的任何回應碼。 報告條件：有非零值 統計資料：最實用的統計數量是 Sum。Minimum、Maximum 和 Average 都會傳回 1。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
HTTPCode_ELB_500_Count	源自於負載平衡器的 HTTP 500 錯誤碼數目。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_502_Count	源自於負載平衡器的 HTTP 502 錯誤碼數目。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
HTTPCode_ELB_503_Count	源自於負載平衡器的 HTTP 503 錯誤碼數目。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
HTTPCode_ELB_504_Count	源自於負載平衡器的 HTTP 504 錯誤碼數目。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
IPv6ProcessedBytes	負載平衡器透過 IPv6 所處理的位元組總數。此計數包含在 ProcessedBytes 中。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
IPv6RequestCount	負載平衡器收到的 IPv6 要求數目。 報告條件：有非零值 統計資料：最實用的統計數量是 Sum。Minimum、Maximum 和 Average 都會傳回 1。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
LowReputationPacketsDropped	從已知惡意來源捨棄的封包數量。當資源層級 DDoS 保護封鎖請求時，會記錄此指標。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
LowReputationRequestsDenied	HTTP 403 回應拒絕的 HTTP 請求數量。當資源層級 DDoS 保護封鎖請求時，會記錄此指標。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
NewConnectionCount	從用戶端到負載平衡器以及從負載平衡器到目標建立的新 TCP 連線總數。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
NonStickyRequestCount	負載平衡器因為無法使用現有的黏性工作階段而選擇新目標時的請求數目。例如，請求是來自新用戶端的第一個請求且黏性 Cookie 不存在、黏性 Cookie 存在但未指定已向此目標群組註冊的目標、黏性 Cookie 的格式不正確或過期，或內部錯誤使負載平衡器無法讀取黏性 Cookie。 報告條件：目標群組上啟用黏性。 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
ProcessedBytes	負載平衡器透過 IPv4 與 IPv6 (HTTP 標頭和 HTTP 承載) 所處理的位元組總數。此計數包括往返用戶端和 Lambda 函數的流量、透過 Websocket 連線的流量，以及啟用使用者身分驗證時來自身分提供者 (IdP) 的流量。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
RejectedConnectionCount	因負載平衡器已達其連線數目上限而拒絕的連線數目。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
RequestCount	透過 IPv4 與 IPv6 處理的要求數目。對於負載平衡器節點能夠選擇目標的請求，此指標的值才會遞增。在選擇目標之前遭拒絕的請求不會反映在此指標中。 報告條件：如果有已註冊的目標，則報告。 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • LoadBalancer ，AvailabilityZone • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup
RuleEvaluations	處理請求時，負載平衡器評估的規則數目。預設規則不會計算在內。此計數包含每個請求的 10 個免費規則評估。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer

AWS/ApplicationELB 命名空間包含負載平衡器容量單位 (LCU) 的下列指標。

指標	Description
ConsumedLCUs	負載平衡器所使用的負載平衡器容量單位 (LCU) 數目。您需要按每小時使用的 LCU 數目付費。當 LCU 保留作用中時，0ConsumedLCUs 將報告，如果0用量超過預留 LCUs，則報告上述值。如需詳細資訊，請參閱 Elastic Load Balancing 定價。 報告條件：一律報告 統計資訊：全部 維度 • LoadBalancer
PeakLCUs	負載平衡器在特定時間點使用的負載平衡器容量單位 (LCU) 數目上限。僅適用於使用 LCU 預留時。 報告條件：永遠 統計資訊：最實用的統計資訊是 Sum 與 Max。 維度 • LoadBalancer
ReservedLCUs	每分鐘報告預留容量的計費指標。在任何期間的總 ReservedLCUs 是您需支付的 LCUs數量。例如，如果保留 500 LCUs 一小時，則每分鐘指標將為 8.33 LCUs。如需詳細資訊，請參閱監控保留。 報告條件：有非零值 統計資訊：全部 維度 • LoadBalancer

AWS/ApplicationELB 命名空間包含下列目標指標。

指標	Description
AnomalousHostCount	偵測到異常的主機數量。 報告條件：一律報告 統計資料：唯一有意義的統計資料是 Minimum和 Maximum。 維度 • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
HealthyHostCount	視為健康的目標數目。 報告條件：如果有已註冊的目標，則報告。 統計資訊：最實用的統計資訊是 Average、Minimum 與 Maximum。 維度 • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup
HTTPCode_Target_2XX_Count ，HTTPCode_Target_3XX_Count ，HTTPCode_Target_4XX_Count ，HTTPCode_Target_5XX_Count	目標所產生的 HTTP 回應碼數目。這未包含負載平衡器所產生的任何回應碼。 報告條件：如果有已註冊的目標，則報告。 統計資料：最實用的統計數量是 Sum。Minimum、Maximum 和 Average 都會傳回 1。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer

指標	Description
MitigatedHostCount	緩解的目標數量。 報告條件：一律報告 統計資訊：最實用的統計資訊是 Average、Minimum 與 Maximum。 維度 • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
RequestCountPerTarget	目標群組中每個目標的平均請求計數。您必須使用 TargetGroup 維度指定目標群組。如果目標是 Lambda 函數，則此指標不適用。 此計數會使用目標群組收到的請求總數，除以目標群組中運作狀態良好的目標數目。如果目標群組中沒有運作狀態良好的目標，則會將其除以已註冊目標的總數。 報告條件：一律報告 統計資訊：唯一有效的統計資訊是 Sum。這代表平均值，而不是總和。 維度 • TargetGroup • TargetGroup ，AvailabilityZone • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup

指標	Description
TargetConnectionErrorCount	負載平衡器與目標之間未成功建立的連線數目。如果目標是 Lambda 函數，則此指標不適用。如果運作狀態檢查連線失敗，此指標不會遞增。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
TargetResponseTime	請求離開負載平衡器後經過的時間，以秒為單位，直到目標開始傳送回應標頭為止。這等同於存取日誌中的 target_processing_time 欄位。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Average 與 pNN.NN (百分位數)。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer

指標	Description
TargetTLSNegotiationErrorCount	未與目標建立工作階段之負載平衡器所啟動的 TLS 連線數目。可能的原因包含晶片或協定不相符。如果目標是 Lambda 函數，則此指標不適用。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer • TargetGroup ，LoadBalancer • TargetGroup ，AvailabilityZone ，LoadBalancer
UnHealthyHostCount	視為不健康的目標數目。 當您取消註冊目標時，這會減少HealthyHostCount 但不會增加 UnhealthyHostCount 。 報告條件：如果有已註冊的目標，則報告。 統計資訊：最實用的統計資訊是 Average、Minimum 與 Maximum。 維度 • LoadBalancer ，TargetGroup • LoadBalancer ，AvailabilityZone ，TargetGroup

指標	Description
ZonalShiftedHostCount	由於區域轉移而被視為已停用的目標數量。 報告條件：有值時報告 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer ，TargetGroup . • AvailabilityZone ，LoadBalancer ，TargetGroup .

AWS/ApplicationELB 命名空間包含下列目標群組運作狀態的指標。如需詳細資訊，請參閱the section called “目標群組運作狀態”。

指標	Description
HealthyStateDNS	符合 DNS 運作狀態良好需求的區域數目。 統計資訊：最實用的統計資訊是 Max。 維度 • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup
HealthyStateRouting	符合路由運作狀態良好需求的區域數目。 統計資訊：最實用的統計資訊是 Max。 維度 • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup
UnhealthyRoutingRequestCount	使用路由容錯移轉動作 (故障開啟) 路由的請求數量。 統計資訊：最實用的統計資訊是 Sum。

指標	Description
	維度 • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup
UnhealthyStateDNS	因不符合 DNS 運作狀態良好需求而在 DNS 中標記為運作狀態不佳的區域數量。 統計資訊：最實用的統計資訊是 Min。 維度 • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup
UnhealthyStateRouting	因不符合路由運作狀態良好需求而導致負載平衡器將流量分散給區域中的所有目標 (包括運作狀態不佳的目標) 的區域數量。 統計資訊：最實用的統計資訊是 Min。 維度 • LoadBalancer ，TargetGroup • AvailabilityZone ，LoadBalancer ，TargetGroup

對於註冊為目標的 Lambda 函數，AWS/ApplicationELB 命名空間包含下列指標。

指標	Description
LambdaInternalError	因為負載平衡器或 AWS Lambda的內部問題而失敗的 Lambda 函數請求數。若要取得錯誤原因代碼，請查看存取日誌的 error_reason 欄位。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。

指標	Description
	維度 • TargetGroup • TargetGroup ，LoadBalancer
LambdaTargetProcessedBytes	負載平衡器針對 Lambda 函數的請求和回應所處理的位元組總數。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer
LambdaUserError	Lambda 函數因為 Lambda 函數有問題而失敗的請求數。例如，負載平衡器沒有叫用函數的許可、負載平衡器從函數收到的 JSON 格式不正確或遺漏必要欄位，或請求內文或回應的大小超過大小上限 1 MB。若要取得錯誤原因代碼，請查看存取日誌的 error_reason 欄位。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • TargetGroup • TargetGroup ，LoadBalancer

AWS/ApplicationELB 命名空間包含下列使用者身分驗證指標。

指標	Description
ELBAuthError	由於身分驗證動作設定錯誤、負載平衡器無法與 IdP 建立連線，或負載平衡器由於內部錯誤而無法完成身分驗證流程，因而無法完成的使用者身分驗證數量。若要取得錯誤原因代碼，請查看存取日誌的 error_reason 欄位。

指標	Description
	報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
ELBAuthFailure	由於 IdP 拒絕使用者存取或授權碼使用多次，因而無法完成的使用者身分驗證數量。若要取得錯誤原因代碼，請查看存取日誌的 error_reason 欄位。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
ELBAuthLatency	向 IdP 查詢 ID 字符和使用者資訊所經歷的時間 (毫秒)。如果其中一或多項操作失敗，此為失敗的時間。 報告條件：有非零值 統計資訊：所有統計資訊都有意義。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
ELBAuthRefreshTokenSuccess	負載平衡器使用 IdP 提供的重新整理字符而成功重新整理使用者宣告的次數。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone , LoadBalancer
ELBAuthSuccess	成功的身分驗證動作次數。此指標在身分驗證工作流程結束時、負載平衡器從 IdP 擷取到使用者宣告之後遞增。 報告條件：有非零值 統計資訊：最實用的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone , LoadBalancer
ELBAuthUserClaimsSizeExceeded	已設定的 IdP 傳回的使用者宣告大小超過 11K 位元組的次數。 報告條件：有非零值 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone , LoadBalancer

AWS/ApplicationELB 命名空間包含目標最佳化工具的下列指標。

指標	Description
TargetControlRequestCount	ALB 轉送給客服人員的請求數量。 報告條件：目標群組上已啟用目標最佳化工具，且有非零值。 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlRequestRejectCount	由於沒有目標準備好接收請求，ALB 拒絕的請求數量。當 TargetControlWorkQueueLength 為零時，此指標會顯示上升。 報告條件：目標群組上已啟用目標最佳化工具，且有非零值。 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlActiveChannelCount	ALB 和客服人員之間的作用中控制頻道數量。對於負載平衡器，這應該等於代理程式的數量。低於預期的數字表示客服人員未正確設定或無法使用。 報告條件：在目標群組上啟用目標最佳化工具，且有非零值。 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlNewChannelCount	在 ALB 和客服人員之間建立的新控制頻道數量。當已安裝代理程式的新目標成功新增至目標群組時，您會在此指標中看到上升。

指標	Description
	報告條件：目標群組上已啟用目標最佳化工具，且具有非零值。 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlChannelErrorCount	ALB 與客服人員之間無法建立或遇到非預期錯誤的控制通道數量。控制頻道錯誤會導致該代理程式 （和目標） 無法接收任何應用程式流量。 報告條件：目標群組上已啟用目標最佳化工具，且有非零值。 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer
TargetControlWorkQueueLength	ALB 從請求請求的客服人員收到的訊號數量。 此資料來自每隔 1 分鐘拍攝的快照。不會擷取次分鐘的變更。 報告條件：目標群組上已啟用目標最佳化工具，且有非零值。 統計資訊：唯一有意義的統計資訊是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

指標	Description
TargetControlProcessedBytes	對於啟用目標最佳化工具的目標群組流量，ALB 處理的位元組數。 報告條件：目標群組上已啟用目標最佳化工具，且有非零值。 統計資料：最有意義的統計資料是 Sum。 維度 • LoadBalancer • AvailabilityZone ，LoadBalancer

Application Load Balancer 的指標維度

若要篩選 Application Load Balancer 的指標，請使用下列維度。

維度	Description
AvailabilityZone	依可用區域篩選指標資料。
LoadBalancer	依負載平衡器篩選指標資料。如下指定負載平衡器：app/load-balancer-name/1234567890123456 (負載平衡器 ARN 的最終部分)。
TargetGroup	依目標群組篩選指標資料。如下指定目標群組：targetgroup/target-group-name/1234567890123456 (目標群組 ARN 的最終部分)。

Application Load Balancer 指標的統計資料

CloudWatch 根據由 Elastic Load Balancing 發佈的指標資料點提供統計資料。統計資料是隨著指定期間的指標資料彙總。當您請求統計資料時，傳回的資料流是藉由指標名稱和維度做識別。維度是可唯一識別指標的名稱/值組。例如，您可以為所有在特定可用區域內啟動的負載平衡器後方之運作狀態良好的 EC2 執行個體請求統計資料。

Minimum 和 Maximum 統計資料會反映每個抽樣時段中個別負載平衡器節點報告的資料點最小和最大值。例如，假設 Application Load Balancer 由 2 個負載平衡器節點組成。一個節點有內含 Minimum

2、Maximum 10、Average 6 的 HealthyHostCount，而其他節點有內含 Minimum 1、Maximum 5、以及 Average 3 的 HealthyHostCount。因此，負載平衡器有 Minimum 1、Maximum 10、以及因為約為 4 的 Average。

我們建議您監控 Minimum 統計資料中的非零值 UnHealthyHostCount，多個資料點出現非零值時提供警示。使用 Minimum 將偵測負載平衡器每個節點和可用區域何時將目標視為運作狀態不佳。如果您想要收到潛在問題的警示，則 Average 或 Maximum 警示非常有用，我們建議客戶檢閱此指標，並在發生次數不為零時進行調查。您可以遵循在 Amazon EC2 Auto Scaling 或 Amazon Elastic Container Service (Amazon ECS) 中使用負載平衡器運作狀態檢查的最佳實務，自動緩解故障。

Sum 統計資料為來自所有負載平衡器節點的彙總值。因為指標包和各期間的多個報告，Sum 僅可用於來自所有負載平衡器節點的彙總指標。

SampleCount 統計資料為測量而得的範本數量。因指標根據範本間隔與事件蒐集而得，此統計資料通常沒有幫助。例如，使用 HealthyHostCount，SampleCount 是根據每個負載平衡器節點回報的範本數量，而非運作狀態良好的主機數量。

百分位數指出資料集之某個值的相對位置。您可以指定任何百分位數，最多使用兩位小數 (例如，p95.45)。例如，第 95 個百分位數表示 95% 的資料低於這個值，而 5% 高於這個值。百分位數通常用於隔離異常。例如，假設應用程式以 1-2 毫秒處理快取中的大部分請求，但如果快取是空的，則是 100-200 毫秒。上限會反映最慢的情況，大約 200 毫秒。平均數不表示資料的分佈。百分位數以更有意義的觀點表達應用程式的效能。您可以使用第 99 個百分位數做為 Auto Scaling 觸發或 CloudWatch 警示，將目標訂為處理時間超過 2 毫秒的請求不超過 1%。

檢視負載平衡器的 CloudWatch 指標

您可以使用 Amazon EC2 主控台來檢視負載平衡器的 CloudWatch 指標。這些指標會以監控圖表的形式顯示。若啟用負載平衡器並接收請求，監控圖表會顯示資料點。

或者，您可以使用 CloudWatch 主控台來檢視負載平衡器的指標。

使用 主控台檢視指標

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 若要檢視由目標群組篩選的指標，請執行下列動作：
   a. 在導覽窗格中，選擇 Target Groups (目標群組)。
   b. 選取您的目標群組，然後選擇 Monitoring (監控) 標籤。
   c. (選用) 若要根據時間篩選結果，請選擇來自 Showing data for (顯示資料) 的時間範圍。

d. 若要放大檢視單一指標，請選取它的圖形。

3. 若要檢視由負載平衡器篩選的指標，請執行下列動作：

a. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。

b. 選取您的負載平衡器，然後選擇 Monitoring (監控) 標籤。

c. (選用) 若要根據時間篩選結果，請選擇來自 Showing data for (顯示資料) 的時間範圍。

d. 若要放大檢視單一指標，請選取它的圖形。

使用 CloudWatch 主控台檢視指標

1. 在 https://console.aws.amazon.com/cloudwatch/ 開啟 CloudWatch 主控台。
2. 在導覽窗格中，選擇指標。
3. 選取 ApplicationELB 命名空間。
4. (選用) 若要檢視所有維度的指標，請在搜尋欄位中輸入其名稱。
5. (選用) 若要根據維度來篩選，請選取下列其中一項：

- 若只要顯示針對負載平衡器而報告的指標，請選擇 Per AppELB Metrics (每個 AppELB 指標)。若要檢視單一負載平衡器的指標，請在搜尋欄位中輸入其名稱。
- 若只要顯示針對目標群組而報告的指標，請選擇 Per AppELB, per TG Metrics (每個 AppELB、每個 TG 指標)。若要檢視單一目標群組的指標，請在搜尋欄位中輸入其名稱。
- 若要依可用區域來只顯示針對負載平衡器而報告的指標，請選擇 Per AppELB, per AZ Metrics (每個 AppELB、每個 AZ 指標)。若要檢視單一負載平衡器的指標，請在搜尋欄位中輸入其名稱。若要檢視單一可用區域的指標，請在搜尋欄位中輸入其名稱。
- 若要依可用區域和目標群組來只顯示針對負載平衡器而報告的指標，請選擇 Per AppELB, per AZ, per TG Metrics (每個 AppELB、每個 AZ、每個 TG 指標)。若要檢視單一負載平衡器的指標，請在搜尋欄位中輸入其名稱。若要檢視單一目標群組的指標，請在搜尋欄位中輸入其名稱。若要檢視單一可用區域的指標，請在搜尋欄位中輸入其名稱。

使用 檢視指標 AWS CLI

使用下列 list-metrics 命令來列出可用指標：

```
aws cloudwatch list-metrics --namespace AWS/ApplicationELB
```

使用 取得指標的統計資料 AWS CLI

使用下列 get-metric-statistics 指令來獲得指定指標與維度的統計資料。CloudWatch 會將不同的維度組合視為不同指標。您無法使用未具體發佈的維度組合來擷取統計資料。您必須指定建立指標時所使用的相同維度。

```
aws cloudwatch get-metric-statistics --namespace AWS/ApplicationELB \
--metric-name UnHealthyHostCount --statistics Average  --period 3600 \
--dimensions Name=LoadBalancer,Value=app/my-load-balancer/50dc6c495c0c9188 \
Name=TargetGroup,Value=targetgroup/my-targets/73e2d6bc24d8a067 \
--start-time 2016-04-18T00:00:00Z --end-time 2016-04-21T00:00:00Z
```

下列為範例輸出：

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-04-18T22:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2016-04-18T04:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        ...
    ],
    "Label": "UnHealthyHostCount"
}
```

Application Load Balancer 的存取日誌

Elastic Load Balancing 提供存取日誌，可針對傳送到負載平衡器的請求，擷取其詳細資訊。每個日誌包含收到請求的時間、用戶端的 IP 地址、延遲、請求路徑和伺服器回應等資訊。您可以使用這些存取日誌來分析流量模式和排除問題。

存取日誌是 Elastic Load Balancing 的選用功能，預設為停用。為負載平衡器啟用存取日誌之後，Elastic Load Balancing 會擷取日誌，並將其以壓縮檔案存放在您指定的 Amazon S3 儲存貯體中。您可以隨時停用存取日誌。

您將需支付 Amazon S3 的儲存費用，但 Elastic Load Balancing 將日誌檔傳送到 Amazon S3 所使用的頻寬不需要付費。如需有關儲存費用的詳細資訊，請參閱 Amazon S3 定價。

目錄

- 存取日誌檔
- 存取日誌項目
- 範例日誌項目
- 設定日誌交付通知
- 處理存取日誌檔
- 為 Application Load Balancer 啟用存取日誌
- 停用 Application Load Balancer 的存取日誌

存取日誌檔

Elastic Load Balancing 每 5 分鐘發佈每個負載平衡器節點的日誌檔。日誌傳遞最終會達到一致。負載平衡器可能在相同期間傳遞多個日誌。這通常是在網站的流量很高時才會發生。

存取日誌的檔案名稱使用以下格式：

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-address_random-string.log.gz
```

bucket

S3 儲存貯體的名稱。

prefix

(選用) 儲存貯體的字首 (邏輯階層)。您指定的字首不得包含字串 AWSLogs。如需詳細資訊，請參閱使用字首組織物件。

AWSLogs

我們在您指定的儲存貯體名稱和可選字首之後，增加了以 AWSLogs 開頭的檔案名稱部分。

aws-account-id

擁有者 AWS 的帳戶 ID。

region

負載平衡器和 S3 儲存貯體的區域。

yyyy/mm/dd

傳遞日誌的日期。

load-balancer-id

負載平衡器的資源 ID。如果資源 ID 包含任何斜線 (/)，斜線會換成句點 (.)。

end-time

記錄間隔結束的日期和時間。例如，結束時間 20140215T2340Z 的日子檔案包含在 23:35 和 23:40 (UTC 或 Zulu 時間) 之間發出的請求項目。

ip-address

處理請求之負載平衡器節點的 IP 地址。對於內部負載平衡器，這是私有 IP 地址。

random-string

系統產生的隨機字串。

以下是含字首的日誌檔案名稱範例：

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-
east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

以下是不含字首的日誌檔案名稱範例：

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/
us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

日誌檔案可存放於儲存貯體任意長時間，但您也可以定義 Amazon S3 生命週期規則，自動封存或刪除日誌檔案。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的物件生命週期管理。

存取日誌項目

Elastic Load Balancing 會記錄向負載平衡器傳送的請求，包括從未送達目標的請求。例如，如果用戶端傳送格式不正確的請求，或沒有運作狀態良好的目標可回應請求，則仍然會記錄此請求。

每個日誌項目包含對負載平衡器提出的單一請求 (或就 WebSocket 而言是連線) 的詳細資訊。對於 WebSocket，只有在連線關閉之後才會寫入項目。如果無法建立升級連線，則項目與 HTTP 或 HTTPS 請求的項目相同。

> **Important**
>
> Elastic Load Balancing 會盡可能記錄請求。建議您使用存取日誌來了解請求的性質，而不是為了全面解釋所有請求。

目錄

- 語法
- 採取的動作
- 分類原因
- 錯誤原因代碼
- 轉換狀態碼

語法

下表依序說明存取日誌項目的欄位。所有欄位以空格分隔。當我們新增欄位時，會將它新增至日誌項目的結尾。當我們準備發佈新欄位時，您可能會在欄位發佈之前看到額外的結尾 "-"。請確定您將日誌剖析設定為在上次記錄欄位後停止，並在我們發佈新欄位後更新日誌剖析。

欄位 （位置）	Description
類型 (1)	請求或連線的類型。可能的值如下所示 (忽略任何其他值)： • `http` – HTTP • `https` – 透過 TLS 傳輸的 HTTP • `h2` – 透過 TLS 傳輸的 HTTP/2 • `grpcs` – 透過 TLS 傳輸的 gRPC • `ws` – WebSocket • `wss` – 透過 TLS 傳輸的 WebSocket

欄位 （位置）	Description
時間 (2)	負載平衡器產生回應給用戶端的時間 (ISO 8601 格式)。對於 WebSocket ，這是連線關閉的時間。
elb (3)	負載平衡器的資源 ID。如果您剖析存取日誌項目，請注意，資源 ID 可能包含斜線 (/)。
client：port (4)	提出請求之用戶端的 IP 地址和連接埠。如果負載平衡器前面有代理，則此欄位會包含代理的 IP 地址。
target：port (5)	處理此請求之目標的 IP 地址和連接埠。 如果用戶端未傳送完整的請求，則負載平衡器無法將請求分派給目標，而這個值會設為 -。 如果目標是 Lambda 函數，這個值會設定為 -。 如果請求被 封鎖 AWS WAF，此值會設為 -。
request_processing _time (6)	從負載平衡器收到請求到請求傳送到目標為止所經過的總時間 (以秒為單位，精確到毫秒)。 如果負載平衡器無法將請求分派給目標，這個值會設為 -1。如果目標在閒置逾時之前關閉連線，或用戶端傳送格式不正確的請求，就可能發生此情況。 如果在達到 10 秒 TCP 連線逾時之前，無法使用目標建立 TCP 連線，此值也可以設定為 -1。 如果 AWS WAF 已啟用 Application Load Balancer 或目標類型為 Lambda 函數，用戶端傳送 POST 請求所需資料所需的時間會計入 `request_processing_time` 。

欄位 （位置）	Description
target_processing_time (7)	從負載平衡器將請求傳送至目標開始，直到目標開始傳送回應標頭為止，所經過的總時間 (以秒為單位，精確到毫秒)。 如果負載平衡器無法將請求分派給目標，這個值會設為 -1。如果目標在閒置逾時之前關閉連線，或用戶端傳送格式不正確的請求，就可能發生此情況。 如果已註冊的目標在閒置逾時之前沒有回應，這個值也可能設為 -1。 如果 AWS WAF Application Load Balancer 未啟用 ，用戶端傳送 POST 請求所需資料所需的時間會計入 `target_processing_time` 。
response_processing_time (8)	從負載平衡器收到目標的回應標頭開始，直到開始將回應傳送到用戶端為止，所經過的總時間 (以秒為單位，精確到毫秒)。這包括負載平衡器上的佇列時間，以及從負載平衡器到用戶端的連線取得時間。 如果負載平衡器未收到來自目標的回應，則此值會設為 -1。如果目標在閒置逾時之前關閉連線，或用戶端傳送格式不正確的請求，就可能發生此情況。
elb_status_code (9)	負載平衡器產生的回應狀態碼、固定回應規則或區塊動作的 AWS WAF 自訂回應碼。
target_status_code (10)	來自目標的回應狀態碼。只有在對目標建立連線且目標傳送回應之後，才會記錄這個值。否則會設為 -。
received_bytes (11)	從用戶端 (請求者) 收到的請求大小 (以位元組為單位)。對於 HTTP 請求，這包括標頭。對於 WebSocket，這是在連線上從用戶端收到的位元組總數。
sent_bytes (12)	傳回到用戶端 (請求者) 的回應大小 (以位元組為單位)。對於 HTTP 請求，這包括回應標頭和內文。對於 WebSocket，這是在連線上傳送到用戶端的位元組總數。 TCP 標頭和 TLS 交握承載不包含在 中`sent_bytes` 。因此 `sent_bytes` 與 `DataTransfer-Out-Bytes` 不相符 AWS Cost Explorer。

欄位 （位置）	Description
"request_line" (13)	來自用戶端的請求行，以雙引號括住，並採用以下格式來記錄：HTTP 方法 + protocol://host:port/uri+HTTP 版本。記錄請求 URI 時，負載平衡器會依原狀保留用戶端傳送的 URL。它不會為存取日誌檔案設定內容類型。處理此欄位時，請考量用戶端如何傳送 URL。
"user_agent" (14)	User-Agent 字串，識別發出請求的用戶端 (以雙引號括住)。此字串包含一或多個產品識別符，product[/version]。如果字串超過 8 KB，則會截斷。
ssl_cipher (15)	[HTTPS 接聽程式] SSL 加密。如果接聽程式不是 HTTPS 接聽程式，此值會設為 -。
ssl_protocol (16)	[HTTPS 接聽程式] SSL 通訊協定。如果接聽程式不是 HTTPS 接聽程式，此值會設為 -。
target_group_arn (17)	目標群組的 Amazon Resource Name (ARN)。
"trace_id" (18)	X-Amzn-Trace-Id 標頭的內容，以雙引號括住。
"domain_name" (19)	[HTTPS 接聽程式] 在 TLS 交握期間由用戶端提供的 SNI 網域，以雙引號括住。如果用戶端不支援 SNI，或網域不符合憑證而向用戶端出示預設憑證，這個值會設為 -。
"chosen_cert_arn" (20)	[HTTPS 接聽程式] 向用戶端出示的憑證的 ARN，以雙引號括住。如果重複使用工作階段，這個值會設為 `session-reused` 。如果接聽程式不是 HTTPS 接聽程式，此值會設為 -。
matched_rule_priority (21)	符合請求之規則的優先順序值。如果規則符合，則此為 1 到 50,000 的值。如果沒有規則符合且採取預設動作，這個值會設為 0。如果在規則評估期間發生錯誤，則會設為 -1。對於任何其他錯誤，則會設為 -。
request_creation_time (22)	負載平衡器從用戶端收到請求的時間 (ISO 8601 格式)。
"actions_executed" (23)	處理請求時所採取的動作，以雙引號括住。這個值是逗號分隔清單，可以包含 採取的動作 中所述的值。如果未採取任何動作，例如對於格式不正確的請求，這個值會設為 -。

欄位（位置）	Description
"redirect_url" (24)	在 HTTP 回應的位置標頭中，指重新導向目標的 URL，以雙引號括住。如果未採取重新導向動作，這個值會設為 -。
"error_reason" (25)	錯誤原因代碼，以雙引號括住。如果請求失敗，則此為錯誤原因代碼所述的其中一個錯誤代碼。如果採取的動作不含驗證動作，或目標不是 Lambda 函數，此值會設為 -。
"target：port_list" (26)	處理此請求之目標之 IP 地址和連接埠以空格分隔的清單，用雙引號括注。目前，列表可以包含一個項目與其匹配的目標：port field。 如果用戶端未傳送完整的請求，則負載平衡器無法將請求分派給目標，而這個值會設為 -。 如果目標是 Lambda 函數，這個值會設定為 -。 如果請求被 封鎖 AWS WAF，此值會設為 -。
"target_status_code_list" (27)	目標回應以空格分隔的狀態代碼清單，用雙引號括注。目前，此清單可以包含一個項目與其匹配的 target_status_code 欄位。 只有在對目標建立連線且目標傳送回應之後，才會記錄這個值。否則會設為 -。
「分類」(28)	去同步緩解的分類 (以雙引號括住)。如果請求不符合 RFC 7230，可能的值是「可接受」、「不明確」和「嚴重」。 如果請求符合 RFC 7230，則此值會設為 -。
"classification_reason" (29)	分類原因代碼 (以雙引號括住)。如果請求不符合 RFC 7230，這是 分類原因 中所述的其中一個分類代碼。如果請求符合 RFC 7230，則此值會設為 -。
conn_trace_id (30)	連線可追蹤性 ID 是用於識別每個連線的唯一不透明 ID。與用戶端建立連線後，來自此用戶端的後續請求會在其各自的存取日誌項目中包含此 ID。此 ID 可做為外部金鑰，在連線和存取日誌之間建立連結。

欄位 （位置）	Description
"transformed_host" (31)	主機標頭在由主機標頭重寫轉換修改之後，即為主機標頭。如果以下任何一項為 true，則此值設定為 -。 • 未套用轉換 • 轉換失敗 • 主機標頭沒有變更，轉換成功 • 沒有原始主機標頭 （例如 HTTP/1.0 請求）
"transformed_uri" (32)	URL 重寫轉換修改後的 URI。如果以下任何一項為 true，則此值設定為 -。 • 未套用轉換 • 轉換失敗 • URI 沒有變更，轉換成功
"request_transform_status" (33)	重寫轉換的狀態。如果未套用重寫轉換，此值會設為 -。否則，此值是 中所述的其中一個狀態值the section called “轉換狀態碼”。

採取的動作

負載平衡器會將它所採取的動作存放在存取日誌的 actions_executed 欄位中。

- `authenticate` – 負載平衡器驗證工作階段、驗證使用者身分，並將使用者資訊新增至請求標頭 (如規則組態所指定)。
- `fixed-response` – 負載平衡器發出固定回應 (如規則組態所指定)。
- `forward` – 負載平衡器將請求轉送至目標 (如規則組態所指定)。
- `redirect` – 負載平衡器將請求重新導向至另一個 URL (如規則組態所指定)。
- `rewrite` — 負載平衡器會重新旋轉請求 URL，如規則組態所指定。
- `waf` – 負載平衡器將請求轉送至 AWS WAF ，以判斷是否要將請求轉送至目標。如果這是最終動作，則 AWS WAF 決定應拒絕請求。根據預設， 拒絕的請求 AWS WAF 會在 `elb_status_code` 欄位中記錄為「403」。當 AWS WAF 設定為使用自訂回應代碼拒絕請求時， `elb_status_code` 欄位將反映設定的回應代碼。
- `waf-failed` — 負載平衡器嘗試將請求轉送至 AWS WAF，但此程序失敗。

分類原因

如果請求不符合 RFC 7230，負載平衡器會在存取日誌的 classification_reason 欄位中存放下列其中一個代碼。如需詳細資訊，請參閱去同步緩解模式。

Code	Description	分類
AmbiguousUri	要求 URI 包含控制字元。	不明確
BadContentLength	Content-Length 標頭包含無法剖析或非有效數字的值。	嚴重
BadHeader	標頭包含空值字元或歸位字元。	嚴重
BadTransferEncoding	Transfer-Encoding 標頭包含錯誤的值。	嚴重
BadUri	要求 URI 包含空值字元或歸位字元。	嚴重
BadMethod	要求方法格式不正確。	嚴重
BadVersion	要求版本格式不正確。	嚴重
BothTeClPresent	要求同時包含 Transfer-Encoding 標頭和 Content-Length 標頭。	不明確
DuplicateContentLength	多個 Content-Length 標頭的值相同。	不明確
EmptyHeader	標頭空白或標頭列僅含空格。	不明確
GetHeadZeroContentLength	GET 或 HEAD 要求的 Content-Length 標頭值為 0。	可接受
MultipleContentLength	多個 Content-Length 標頭的值不同。	嚴重

Code	Description	分類
`MultipleTransferEncodingChunked`	有多個 Transfer-Encoding：區塊標頭。	嚴重
`NonCompliantHeader`	標頭包含非 ASCII 或控制字元。	可接受
`NonCompliantVersion`	要求版本包含錯誤的值。	可接受
`SpaceInUri`	要求 URI 包含非 URL 編碼的空格。	可接受
`SuspiciousHeader`	可使用通用文字正規化技術將標頭正規化為 Transfer-Encoding 或 Content-Length。	不明確
`SuspiciousTeClPresent`	請求同時包含 Transfer-Encoding 標頭和 Content-Length 標頭，其中至少有一個是可疑的。	嚴重
`UndefinedContentLengthSemantics`	GET 或 HEAD 請求有定義的 Content-Length 標頭。	不明確
`UndefinedTransferEncodingSemantics`	GET 或 HEAD 請求有定義的 Transfer-Encoding 標頭。	不明確

錯誤原因代碼

如果負載平衡器無法完成驗證動作，負載平衡器會將以下其中一個原因代碼存放在存取日誌的 error_reason 欄位。負載平衡器還會遞增對應的 CloudWatch 指標。如需詳細資訊，請參閱使用 Application Load Balancer 來驗證使用者身分。

Code	Description	指標
AuthInvalidCookie	身分驗證 Cookie 無效。	ELBAuthFailure
AuthInvalidGrantError	來自字符端點的授權碼無效。	ELBAuthFailure
AuthInvalidIdToken	ID 字符無效。	ELBAuthFailure
AuthInvalidStateParam	state 參數無效。	ELBAuthFailure
AuthInvalidTokenResponse	來自字符端點的回應無效。	ELBAuthFailure
AuthInvalidUserinfoResponse	來自使用者資訊端點的回應無效。	ELBAuthFailure
AuthMissingCodeParam	來自授權端點的身分驗證回應缺少名為 'code' 的查詢參數。	ELBAuthFailure
AuthMissingHostHeader	來自授權端點的身分驗證回應缺少主機標頭欄位。	ELBAuthError
AuthMissingStateParam	來自授權端點的身分驗證回應缺少名為 'state' 的查詢參數。	ELBAuthFailure
AuthTokenEpRequestFailed	字符端點傳回錯誤回應 (非 2XX)。	ELBAuthError
AuthTokenEpRequestTimeout	負載平衡器無法與權杖端點通訊，或權杖端點未在 5 秒內回應。	ELBAuthError

Code	Description	指標
AuthUnhandledException	負載平衡器發生未處理的例外狀況。	ELBAuthError
AuthUserinfoEpRequestFailed	IdP 使用者資訊端點傳回錯誤回應 (非 2XX)。	ELBAuthError
AuthUserinfoEpRequestTimeout	負載平衡器無法與 IdP 使用者資訊端點通訊，或使用者資訊端點未在 5 秒內回應。	ELBAuthError
AuthUserinfoResponseSizeExceeded	IdP 傳回的宣告大小超過 11K 位元組。	ELBAuthUserClaimsSizeExceeded

如果負載平衡器無法完成 jwt-validation 動作，負載平衡器會將下列其中一個原因碼儲存在存取日誌的 error_reason 欄位中。負載平衡器還會遞增對應的 CloudWatch 指標。如需詳細資訊，請參閱使用 Application Load Balancer 驗證 JWTs。

Code	Description	指標
JWTHeaderNotPresent	請求不包含授權標頭。	JWTValidationFailureCount
JWTRequestFormatInvalid	請求中的權杖格式錯誤或缺少必要部分（標頭、承載或簽章），標頭不包含「承載」字首，標頭包含不同的身分驗證類型，例如「基本」，授權標頭存在，但權杖不存在，如果請求中存在多個權杖	JWTValidationFailureCount
JWKSRequestTimeout	負載平衡器無法與 JWKS 端點通訊，或 JWKS 端點未在 5 秒內回應。	JWTValidationFailureCount

Code	Description	指標
JWKSResponseSizeExceeded	JWKS 端點傳回的回應大小超過 150KB，或 JWKS 端點傳回的金鑰數目超過 10。	JWTValidationFailureCount
JWKSRequestFailed	JWKS 端點有錯誤回應 （非 2XX)。	JWTValidationFailureCount
JWKSResponseInvalid	JWKS 回應有下列一或多個問題：非 JSON 格式、無效的字元、無效的 JWKS 格式、缺少/無效的強制性 JWKS 屬性、公有金鑰具有不支援的演算法、公有金鑰無法轉換為解碼金鑰、公有金鑰大小不是 2K。	JWTValidationFailureCount
JWTSignatureValidationError	無法基於任何原因驗證字符簽章，包括簽章不相符、字符使用不支援的演算法簽署、字符中的 KID 不存在於 JWKS 端點中。	JWTValidationFailureCount
JWTClaimNotPresent	用戶端請求中的 JWT 不包含驗證所需的宣告	JWTValidationFailureCount
JWTClaimFormatInvalid	JWT 中宣告值的格式與組態中指定的格式不相符	JWTValidationFailureCount
JWTClaimValueInvalid	JWT 中宣告的值無效。	JWTValidationFailureCount
JWTValidationInternalError	在用戶端請求中驗證 JWT 時，負載平衡器遇到意外錯誤。	JWTValidationFailureCount

如果對權重目標群組的請求失敗，負載平衡器會將以下其中一個錯誤代碼存放在存取日誌的 error_reason 欄位。

Code	Description
AWSALBTGCookieInvalid	與加權目標群組一起使用的 AWSALBTG Cookie 無效。例如，當 Cookie 值是 URL 編碼時，負載平衡器會傳回此錯誤。
WeightedTargetGroupsUnhandledException	負載平衡器發生未處理的例外狀況。

如果對 Lambda 函數的請求失敗，負載平衡器會將以下其中一個原因代碼存放在存取日誌的 error_reason 欄位。負載平衡器還會遞增對應的 CloudWatch 指標。如需詳細資訊，請參閱 Lambda Invoke 動作。

Code	Description	指標
LambdaAccessDenied	負載平衡器沒有叫用 Lambda 函數的許可。	LambdaUserError
LambdaBadRequest	Lambda 呼叫失敗，因為用戶端要求標頭或內文不只包含 UTF-8 字元。	LambdaUserError
LambdaConnectionError	負載平衡器無法連線到 Lambda。	LambdaInternalError
LambdaConnectionTimeout	嘗試連接到 Lambda 逾時。	LambdaInternalError
LambdaEC2AccessDeniedException	Amazon EC2 在函數初始化期間拒絕存取 Lambda。	LambdaUserError
LambdaEC2ThrottledException	Amazon EC2 在函數初始化期間對 Lambda 進行節流。	LambdaUserError

Code	Description	指標
LambdaEC2UnexpectedException	Amazon EC2 在函數初始化期間發生意外例外狀況。	LambdaUserError
LambdaENILimitReachedException	Lambda 無法在 Lambda 函數組態所指定的 VPC 中建立網路介面，因為已超出網路介面數量限制。	LambdaUserError
LambdaInvalidResponse	來自 Lambda 函數的回應格式不正確或缺少所需的欄位。	LambdaUserError
LambdaInvalidRuntimeException	不支援指定的 Lambda 執行期版本。	LambdaUserError
LambdaInvalidSecurityGroupIDException	Lambda 函數之組態中指定的安全群組 ID 無效。	LambdaUserError
LambdaInvalidSubnetIDException	Lambda 函數的組態中指定的子網路 ID 無效。	LambdaUserError
LambdaInvalidZipFileException	Lambda 無法解壓縮指定的函數 zip 檔案。	LambdaUserError
LambdaKMSAccessDeniedException	Lambda 無法解密環境變數，因為對 KMS 金鑰的存取遭拒。請檢查 Lambda 函數的 KMS 許可。	LambdaUserError
LambdaKMSDisabledException	Lambda 無法解密環境變數，因為指定的 KMS 金鑰已停用。請檢查 Lambda 函數的 KMS 金鑰設定。	LambdaUserError

Code	Description	指標
LambdaKMSInvalidStateException	Lambda 無法解密環境變數，因為 KMS 金鑰的狀態無效。請檢查 Lambda 函數的 KMS 金鑰設定。	LambdaUserError
LambdaKMSNotFoundException	Lambda 無法解密環境變數，因為找不到 KMS 金鑰。請檢查 Lambda 函數的 KMS 金鑰設定。	LambdaUserError
LambdaRequestTooLarge	請求本文的大小超過 1 MB。	LambdaUserError
LambdaResourceNotFound	找不到 Lambda 函數。	LambdaUserError
LambdaResponseTooLarge	回應的大小超過 1 MB。	LambdaUserError
LambdaServiceException	Lambda 發生內部錯誤。	LambdaInternalError
LambdaSubnetIPAddressLimitReachedException	Lambda 無法設定 Lambda 函數的 VPC 存取，因為一個或多個子網路沒有可用的 IP 地址。	LambdaUserError
LambdaThrottling	因為有太多請求，Lambda 函數受到節制。	LambdaUserError
LambdaUnhandled	Lambda 函數發生未處理的例外狀況。	LambdaUserError
LambdaUnhandledException	負載平衡器發生未處理的例外狀況。	LambdaInternalError

Code	Description	指標
LambdaWebsocketNotSupported	Lambda 不支援 WebSocket。	LambdaUserError

如果負載平衡器在轉送請求時遇到錯誤 AWS WAF，它會在存取日誌的 error_reason 欄位中存放下列其中一個錯誤代碼。

Code	Description
WAFConnectionError	負載平衡器無法連線至 AWS WAF。
WAFConnectionTimeout	與 的連線 AWS WAF 已逾時。
WAFResponseReadTimeout	AWS WAF 逾時的請求。
WAFServiceError	AWS WAF 傳回 5XX 錯誤。
WAFUnhandledException	負載平衡器發生未處理的例外狀況。

轉換狀態碼

Code	Description
TransformBufferTooSmall	重寫轉換失敗，因為結果超過內部緩衝區的大小。嘗試讓規則表達式較不複雜。
TransformCompileError	規則表達式的編譯失敗。
TransformCompileTooBig	編譯的規則表達式太大。嘗試讓規則表達式較不複雜。

Code	Description
TransformInvalidHost	主機標頭重寫轉換失敗，因為產生的主機無效。
TransformInvalidPath	URL 重寫轉換失敗，因為產生的路徑無效。
TransformRegexSyntaxError	規則表達式包含語法錯誤。
TransformReplaceError	轉換取代失敗。
TransformSuccess	重寫轉換已成功完成。

範例日誌項目

以下為日誌項目範例。請注意，範例文字只會在多行上顯示，以便於閱讀。

範例 HTTP 項目

以下是 HTTP 接聽程式的範例日誌項目 (連接埠 80 到連接埠 80)：

```
http 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337262-36d228ad5d99923122bbe354" "-" "-"
0 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.1:80" "200" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

範例 HTTPS 項目

以下是 HTTPS 接聽程式的範例日誌項目 (連接埠 443 到連接埠 80)：

```
https 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.086 0.048 0.037 200 200 0 57
"GET https://www.example.com:443/ HTTP/1.1" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
 TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
```

```
"Root=1-58337281-1d84f3d73c47ec4e58577259" "www.example.com" "arn:aws:acm:us-
east-2:123456789012:certificate/12345678-1234-1234-1234-123456789012"
1 2018-07-02T22:22:48.364000Z "authenticate,forward" "-" "-" "10.0.0.1:80" "200" "-"
 "-"
TID_1234abcd5678ef90 "m.example.com" "-" "TransformSuccess"
```

HTTP/2 項目範例

以下是 HTTP/2 串流的範例日誌項目。

```
h2 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.1.252:48160 10.0.0.66:9000 0.000 0.002 0.000 200 200 5 257
"GET https://10.0.2.105:773/ HTTP/2.0" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
 TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337327-72bd00b0343d75b906739c42" "-" "-"
1 2018-07-02T22:22:48.364000Z "redirect" "https://example.com:80/" "-" "10.0.0.66:9000"
 "200" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

WebSocket 項目範例

以下是 WebSocket 連線的範例日誌項目。

```
ws 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.0.140:40914 10.0.1.192:8010 0.001 0.003 0.000 101 101 218 587
"GET http://10.0.0.30:80/ HTTP/1.1" "-" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.1.192:8010" "101" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

範例安全 WebSocket 項目

以下是安全 WebSocket 連線的範例日誌項目。

```
wss 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.0.140:44244 10.0.0.171:8010 0.000 0.001 0.000 101 101 218 786
"GET https://10.0.0.30:443/ HTTP/1.1" "-" ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2
```

```
arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.171:8010" "101" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

Lambda 函數的範例項目

以下是對 Lambda 函數之請求成功的範例日誌項目：

```
http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
0 2018-11-30T22:22:48.364000Z "forward" "-" "-" "-" "-" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

以下是對 Lambda 函數之請求失敗的範例日誌項目：

```
http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 502 - 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
0 2018-11-30T22:22:48.364000Z "forward" "-" "LambdaInvalidResponse" "-" "-" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

設定日誌交付通知

若要在 Elastic Load Balancing 將日誌交付至 S3 儲存貯體時接收通知，請使用 Amazon S3 事件通知。Elastic Load Balancing 使用 PutObject、CreateMultipartUpload 和 POST 物件將日誌交付至 Amazon S3。為了確保您收到所有日誌交付通知，請在組態中包含所有這些物件建立事件。

如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 Amazon S3 事件通知。

處理存取日誌檔

存取日誌檔已壓縮。如果您下載檔案，則必須先將其解壓縮才能看到資訊。

如果您的網站上有許多需求，負載平衡器產生的日誌檔可能有好幾 GB 的資料。您可能無法逐行處理這麼龐大的資料。因此，您可能需要使用提供平行處理解決方案的分析工具。例如，您可以使用以下分析工具來分析和處理存取日誌：

- Amazon Athena 是一種互動式查詢服務，可讓您使用標準 SQL 輕鬆分析 Amazon S3 中的資料。如需詳細資訊，請參閱《Amazon Athena 使用者指南》中的查詢 Application Load Balancer 日誌。
- Loggly
- Splunk
- Sumo Logic

為 Application Load Balancer 啟用存取日誌

為負載平衡器啟用存取記錄時，必須指定供負載平衡器存放日誌的 S3 儲存貯體名稱。儲存貯體必須具有儲存貯體政策，能授予 Elastic Load Balancing 寫入儲存貯體的許可。

任務

- 步驟 1：建立 S3 儲存貯體
- 步驟 2：連接政策到您的 S3 儲存貯體
- 步驟 3：設定存取日誌
- 步驟 4：確認儲存貯體許可
- 疑難排解

步驟 1：建立 S3 儲存貯體

啟用存取日誌時，必須為存取日誌指定 S3 儲存貯體。您可以使用現有儲存貯體，也可以建立專門用於存取日誌的儲存貯體。儲存貯體必須符合下列需求。

要求

- 儲存貯體與負載平衡器必須位於相同的 Region (區域)。儲存貯體和負載平衡器可以由不同的帳戶擁有。
- Amazon S3 受管金鑰 (SSE-S3) 是唯一支援的伺服器端加密選項。如需詳細資訊，請參閱 Amazon S3 受管加密金鑰 (SSE-S3)。

使用 Amazon S3 主控台建立 S3 儲存貯體

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選擇建立儲存貯體。
3. 在 Create bucket (建立儲存貯體) 頁面上，執行下列操作：
   a. 針對 Bucket name (儲存貯體名稱)，輸入儲存貯體的名稱。該名稱在 Amazon S3 中所有現有的儲存貯體名稱之間，不得重複。在某些區域，可能會對儲存貯體的名稱進行其他限制。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的儲存貯體限制。
   b. 針對 AWS 區域，選取您建立負載平衡器時所在的區域。
   c. 對於預設加密，選擇 Amazon S3 受管金鑰 (SSE-S3)。
   d. 選擇建立儲存貯體。

步驟 2：連接政策到您的 S3 儲存貯體

您的 S3 儲存貯體必須擁有儲存貯體政策，以授權 Elastic Load Balancing 將存取日誌寫入到儲存貯體。儲存貯體政策是以存取政策語言所編寫的 JSON 陳述式集合，可定義儲存貯體的存取許可。每個陳述式包含單一許可的相關資訊，且包含一系列的元素。

如果您目前使用的儲存貯體有已連接的政策，您可以將 Elastic Load Balancing 存取日誌的陳述式加入至政策中。若您這麼做，建議您評估所產生的一組許可，以確保它們適用於需要存取儲存貯體以取得存取日誌的使用者。

儲存貯體政策

此政策會將許可授予日誌交付服務。

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
```

```
}
```

針對 Resource，使用範例政策中顯示的格式，輸入存取日誌位置的 ARN。一律在 S3 儲存貯體 ARN 的資源路徑中包含具有負載平衡器的帳戶 ID。這可確保只有來自指定帳戶的負載平衡器才能將存取日誌寫入 S3 儲存貯體。

您指定的 ARN 取決於您是否計劃在步驟 3 中啟用存取日誌時包含字首。

字首為 的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket，字首為 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

AWS GovCloud (US) – 下列範例使用 的 ARN 語法 AWS GovCloud (US) Regions。

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

沒有字首的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket。S3 儲存貯體 ARN 中沒有字首部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

AWS GovCloud (US) – 下列範例使用 的 ARN 語法 AWS GovCloud (US) Regions。

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

舊版儲存貯體政策

先前，對於 2022 年 8 月之前可用的區域，我們需要政策，將許可授予該區域特定的 Elastic Load Balancing 帳戶。仍然支援此舊版政策，但建議您將其取代為上述較新的政策。如果您偏好繼續使用此處未顯示的舊版政策，您可以這麼做。

以下是舊版政策中要在 Principal 中指定的 Elastic Load Balancing 帳戶的 IDs，以供參考。請注意，不在此清單中的區域不支援舊版政策。

- 美國東部 (維吉尼亞北部) – 127311923021

- 美國東部 (俄亥俄) – 033677994240
- 美國西部 (加利佛尼亞北部) – 027434742980
- 美國西部 (奧勒岡) – 797873946194
- 非洲 (開普敦) – 098369216593
- 亞太區域 (香港) – 754344448648
- 亞太區域 (雅加達) – 589379963580
- 亞太區域 (孟買) – 718504428378
- 亞太區域 (大阪) – 383597477331
- 亞太區域 (首爾) – 600734575887
- 亞太區域 (新加坡) – 114774131450
- 亞太區域 (雪梨) – 783225319266
- 亞太區域 (東京) – 582318560864
- 加拿大 (中部) – 985666609251
- 歐洲 (法蘭克福) – 054676820928
- 歐洲 (愛爾蘭) – 156460612806
- 歐洲 (倫敦) – 652711504416
- 歐洲 (米蘭) – 635631232127
- 歐洲 (巴黎) – 009996457667
- 歐洲 (斯德哥爾摩) – 897822967062
- 中東 (巴林) – 076674570225
- 南美洲 (聖保羅) – 507241528517
- AWS GovCloud （美國東部） – 190560391635
- AWS GovCloud （美國西部） – 048591011584

Outpost 區域

以下政策會將許可授予指定的日誌交付服務。將此政策用於 Outpost 區域中的負載平衡器。

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logdelivery.elb.amazonaws.com"
    },
```

```
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*",
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

針對 Resource，使用範例政策中顯示的格式，輸入存取日誌位置的 ARN。一律在 S3 儲存貯體 ARN 的資源路徑中包含具有負載平衡器的帳戶 ID。這可確保只有來自指定帳戶的負載平衡器才能將存取日誌寫入 S3 儲存貯體。

您指定的 S3 儲存貯體 ARN 取決於您是否計劃在步驟 3 中啟用存取日誌時包含字首。

字首為 的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket，字首為 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

沒有字首的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket。S3 儲存貯體 ARN 中沒有字首部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

安全最佳實務

- 使用完整資源路徑，包括 S3 儲存貯體 ARN 的帳戶 ID 部分。請勿在 S3 儲存貯體 ARN 的帳戶 ID 部分中使用萬用字元 (*)。

  ```
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
  ```

- 使用 來aws:SourceArn確保只有來自指定區域和帳戶的負載平衡器才能使用您的儲存貯體。

  ```
  "Condition": {
      "ArnLike": {
          "aws:SourceArn":
   "arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/*"
      }
  ```

```
}
```

- aws:SourceOrgId 搭配 使用，aws:SourceArn以確保只有來自指定組織的負載平衡器才能使用您的儲存貯體。

```
"Condition": {
    "StringEquals": {
        "aws:SourceOrgId": "o-1234567890"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:elasticloadbalancing:*:*:loadbalancer/*"
    }
}
```

- 如果您有Deny陳述式來防止存取服務主體，但明確允許的服務主體除外，請務必將 logdelivery.elasticloadbalancing.amazonaws.com新增至允許的服務主體清單。例如，如果您使用 aws:PrincipalServiceNamesList條件，請新增 logdelivery.elasticloadbalancing.amazonaws.com ，如下所示：

```
{
  "Effect": "Deny",
  "Principal": "*",
  "Condition": {
      "StringNotEqualsIfExists": {
          "aws:PrincipalServiceNamesList": [
              "logdelivery.elasticloadbalancing.amazonaws.com",
              "service.amazonaws.com"
          ]
      }
  }
}
```

如果您使用 NotPrincipal元素，請新增 logdelivery.elasticloadbalancing.amazonaws.com ，如下所示。請注意，我們建議您使用 aws:PrincipalServiceName或 aws:PrincipalServiceNamesList條件金鑰來明確允許服務主體，而不是使用 NotPrincipal元素。如需詳細資訊，請參閱 NotPrincipal。

```
{
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
```

```
        "logdelivery.elasticloadbalancing.amazonaws.com",
        "service.amazonaws.com"
      ]
    }
},
```

建立儲存貯體政策後，請使用 Amazon S3 介面，例如 Amazon S3 主控台或 AWS CLI 命令，將儲存貯體政策連接至 S3 儲存貯體。

Console

將儲存貯體政策連接至 S3 儲存貯體

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選取儲存貯體的名稱，開啟其詳細資訊頁面。
3. 選擇 Permissions (許可)，然後選擇 Bucket policy (儲存貯體政策)、Edit (編輯)。
4. 更新儲存貯體政策，授予所需許可。
5. 選擇儲存變更。

AWS CLI

將儲存貯體政策連接至 S3 儲存貯體

使用 put-bucket-policy 命令。在此範例中，儲存貯體政策已儲存至指定的 .json 檔案。

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://access-log-policy.json
```

步驟 3：設定存取日誌

使用下列程序來設定存取日誌，以擷取請求資訊並將日誌檔案交付至 S3 儲存貯體。

要求

儲存貯體必須符合步驟 1 中所述的要求，且您必須按照步驟 2 所述連接儲存貯體政策。如果您包含字首，則不得包含字串 "AWSLogs"。

管理存取日誌的 S3 儲存貯體

在刪除您為存取日誌設定的儲存貯體之前，請務必停用存取日誌。否則，如果新的儲存貯體有相同名稱，且所需的儲存貯體政策是在您未擁有的 AWS 帳戶 中建立，則 Elastic Load Balancing 可能會將負載平衡器的存取日誌寫入這個新的儲存貯體。

Console

啟用存取日誌

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取您負載平衡器的名稱來開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 對於監控，請開啟存取日誌。
6. 針對 S3 URI，請輸入日誌檔案的 S3 URI。指定的 URI 取決於您是否使用字首。
   - 字首為 s3：//*amzn-s3-demo-logging-bucket*/*logging-prefix* 的 URI
   - 沒有字首的 URI：s3：//*amzn-s3-demo-logging-bucket*
7. 選擇儲存變更。

AWS CLI

啟用存取日誌

使用 modify-load-balancer-attributes 命令搭配相關屬性。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes \
        Key=access_logs.s3.enabled,Value=true \
        Key=access_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \
        Key=access_logs.s3.prefix,Value=logging-prefix
```

CloudFormation

啟用存取日誌

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含相關屬性。

```
Resources:
```

```
myLoadBalancer:
  Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
  Properties:
    Name: my-alb
    Type: application
    Scheme: internal
    Subnets:
      - !Ref subnet-AZ1
      - !Ref subnet-AZ2
    SecurityGroups:
      - !Ref mySecurityGroup
    LoadBalancerAttributes:
      - Key: "access_logs.s3.enabled"
        Value: "true"
      - Key: "access_logs.s3.bucket"
        Value: "amzn-s3-demo-logging-bucket"
      - Key: "access_logs.s3.prefix"
        Value: "logging-prefix"
```

步驟 4：確認儲存貯體許可

為負載平衡器啟用存取日誌之後，Load Balancing 會驗證 S3 儲存貯體，並建立測試檔案，以確保儲存貯體政策指定所需的許可。您可以使用 Amazon S3 主控台來確認是否已建立測試檔案。測試檔案不是實際的存取日誌檔案；它不包含範例記錄。

驗證是否已使用 Amazon S3 主控台在您的儲存貯體中建立測試檔案

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選取您為存取日誌指定的儲存貯體名稱。
3. 導覽到測試檔案，ELBAccessLogTestFile。位置取決於您是否使用字首。
   - 字首為 *amzn-s3-demo-logging-bucket*/*logging-prefix*/AWSLogs/*123456789012*/ELBAccessLogTestFile 的位置
   - 沒有字首的位置：*amzn-s3-demo-logging-bucket*/AWSLogs/*123456789012*/ELBAccessLogTestFile

疑難排解

如果您收到存取遭拒錯誤，則以下是可能的原因：

- 儲存貯體政策不會授權 Elastic Load Balancing 將存取日誌寫入儲存貯體。確認您正在使用適合該區域的正確儲存貯體政策。確認資源 ARN 使用您在啟用存取日誌時指定的相同儲存貯體名稱。如果啟用存取日誌時未指定字首，則請確認資源 ARN 不包含字首。
- 儲存貯體使用不支援的伺服器端加密選項。儲存貯體必須使用 Amazon S3 受管金鑰 (SSE-S3)。

停用 Application Load Balancer 的存取日誌

您可以隨時對負載平衡器停用存取日誌。停用存取日誌之後，存取日誌會保留在 S3 儲存貯體中，直到您刪除為止。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立、設定和使用 S3 儲存貯體。 Amazon S3

Console

停用存取日誌

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取您負載平衡器的名稱來開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 針對監控，請關閉存取日誌。
6. 選擇儲存變更。

AWS CLI

停用存取日誌

使用 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes Key=access_logs.s3.enabled,Value=false
```

Application Load Balancer 的連線日誌

Elastic Load Balancing 提供連線日誌，可擷取傳送至負載平衡器之請求的詳細資訊。每個日誌都包含資訊，例如用戶端的 IP 地址和連接埠、接聽程式連接埠、使用的 TLS 密碼和通訊協定、TLS 交握延遲、連線狀態和用戶端憑證詳細資訊。您可以使用這些連線日誌來分析請求模式並疑難排解問題。

連線日誌是 Elastic Load Balancing 的選用功能，預設為停用。在您啟用負載平衡器的連線日誌後，Elastic Load Balancing 會擷取日誌，並將其存放在您指定的 Amazon S3 儲存貯體中，做為壓縮檔案。您可以隨時停用連線日誌。

您將需支付 Amazon S3 的儲存費用，但 Elastic Load Balancing 將日誌檔傳送到 Amazon S3 所使用的頻寬不需要付費。如需有關儲存費用的詳細資訊，請參閱 Amazon S3 定價。

目錄

- 連線日誌檔案
- 連線日誌項目
- 範例日誌項目
- 處理連線日誌檔案
- 啟用 Application Load Balancer 的連線日誌
- 停用 Application Load Balancer 的連線日誌

連線日誌檔案

Elastic Load Balancing 每 5 分鐘發佈每個負載平衡器節點的日誌檔。日誌傳遞最終會達到一致。負載平衡器可能在相同期間傳遞多個日誌。這通常是在網站的流量很高時才會發生。

連線日誌的檔案名稱使用下列格式：

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/
conn_log_aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-
address_random-string.log.gz
```

bucket

S3 儲存貯體的名稱。

prefix

(選用) 儲存貯體的字首 (邏輯階層)。您指定的字首不得包含字串 AWSLogs。如需詳細資訊，請參閱使用字首組織物件。

AWSLogs

我們在您指定的儲存貯體名稱和可選字首之後，增加了以 AWSLogs 開頭的檔案名稱部分。

aws-account-id

擁有者的帳戶 AWS ID。

region

負載平衡器和 S3 儲存貯體的區域。

yyyy/mm/dd

傳遞日誌的日期。

load-balancer-id

負載平衡器的資源 ID。如果資源 ID 包含任何斜線 (/)，斜線會換成句點 (.)。

end-time

記錄間隔結束的日期和時間。例如，結束時間 20140215T2340Z 的日子檔案包含在 23:35 和 23:40 (UTC 或 Zulu 時間) 之間發出的請求項目。

ip-address

處理請求之負載平衡器節點的 IP 地址。對於內部負載平衡器，這是私有 IP 地址。

random-string

系統產生的隨機字串。

以下是含字首的日誌檔案名稱範例：

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/
conn_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

以下是不含字首的日誌檔案名稱範例：

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/conn_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

日誌檔案可存放於儲存貯體任意長時間，但您也可以定義 Amazon S3 生命週期規則，自動封存或刪除日誌檔案。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的物件生命週期管理。

連線日誌項目

每個連線嘗試在連線日誌檔案中都有一個項目。用戶端請求的傳送方式取決於持續或非持續的連線。非持久性連線具有單一請求，可在存取日誌和連線日誌中建立單一項目。持久性連線有多個請求，可在存取日誌中建立多個項目，並在連線日誌中建立單一項目。

目錄

- 語法
- 錯誤原因代碼

語法

下表依序說明連線日誌項目的欄位。所有欄位以空格分隔。當我們新增欄位時，會將它新增至日誌項目的結尾。當我們準備發佈新欄位時，您可能會在欄位發佈之前看到額外的結尾 "-"。請確定您將日誌剖析設定為在上次記錄的欄位之後停止，並在我們發佈新欄位之後更新日誌剖析。

欄位 （位置）	Description
時間戳記 (1)	負載平衡器成功建立或無法建立連線的時間，採用 ISO 8601 格式。
client_ip (2)	請求用戶端的 IP 地址。
client_port (3)	請求用戶端的連接埠。
listener_port (4)	接收用戶端請求的負載平衡器接聽程式連接埠。
tls_protocol (5)	【HTTPS 接聽程式】 交握期間使用的 SSL/TLS 通訊協定。對於非 SSL/TLS - 請求，此欄位設定為 。
tls_cipher (6)	【HTTPS 接聽程式】 交握期間使用的 SSL/TLS 通訊協定。對於非 SSL/TLS - 請求，此欄位設定為 。

欄位 （位置）	Description
tls_handshake_latency (7)	【HTTPS 接聽程式】 建立成功交握時經過的總秒數，精確度為毫秒。此欄位在下列-情況下設定為 ： • 傳入請求不是 SSL/TLS 請求。 • 未成功建立交握。
leaf_client_cert_subject (8)	【HTTPS 接聽程式】 分葉用戶端憑證的主旨名稱。此欄位在下列-情況下設定為 ： • 傳入請求不是 SSL/TLS 請求。 • 負載平衡器接聽程式未設定為啟用 mTLS。 • 伺服器無法載入/剖析分葉用戶端憑證。
leaf_client_cert_validity (9)	【HTTPS 接聽程式】 分葉用戶端憑證的有效性，使用 `not-after` ISO 8601 格式的 `not-before` 和 。此欄位在下列-情況下設定為 ： • 傳入請求不是 SSL/TLS 請求。 • 負載平衡器接聽程式未設定為啟用 mTLS。 • 伺服器無法載入/剖析分葉用戶端憑證。
leaf_client_cert_serial_number (10)	【HTTPS 接聽程式】 分葉用戶端憑證的序號。此欄位在下列-情況下設定為 ： • 傳入請求不是 SSL/TLS 請求。 • 負載平衡器接聽程式未設定為啟用 mTLS。 • 伺服器無法載入/剖析分葉用戶端憑證。
tls_verify_status (11)	【HTTPS 接聽程式】 連線請求的狀態。`Success` 如果成功建立連線，則此值為 。在連線失敗時，值為 `Failed:$error_code` 。
conn_trace_id (12)	連線可追蹤性 ID 是用於識別每個連線的唯一不透明 ID。與用戶端建立連線後，來自此用戶端的後續請求會在其各自的存取日誌項目中包含此 ID。此 ID 可做為外部金鑰，在連線和存取日誌之間建立連結。
tls_keyexchange (13)	【HTTPS 接聽程式】 TLS 或 PQ-TLS 交握期間使用的金鑰交換。對於非 SSL/TLS - 請求，此欄位設定為 。

錯誤原因代碼

如果負載平衡器無法建立連線，負載平衡器會將下列其中一個原因代碼儲存在連線日誌中。

Code	Description	
ClientCertMaxChainDepthExceeded	已超過用戶端憑證鏈深度上限	
ClientCertMaxSizeExceeded	已超過用戶端憑證大小上限	
ClientCertCrlHit	CA 已撤銷用戶端憑證	
ClientCertCrlProcessingError	CRL 處理錯誤	
ClientCertUntrusted	用戶端憑證不受信任	
ClientCertNotYetValid	用戶端憑證尚未有效	
ClientCertExpired	用戶端憑證已過期	
ClientCertTypeUnsupported	不支援用戶端憑證類型	
ClientCertInvalid	用戶端憑證無效	

Code	Description	
ClientCertPurposeInvalid	用戶端憑證用途無效	
ClientCertRejected	自訂伺服器驗證拒絕用戶端憑證	
UnmappedConnectionError	未映射的執行時間連線錯誤	

範例日誌項目

以下是連線日誌項目的範例。請注意，範例文字只會出現在多行上，使其更容易閱讀。

以下是在連接埠 443 上啟用交互 TLS 驗證模式的 HTTPS 接聽程式成功連線的日誌項目範例。

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
 4.036
"CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Success TID_3180a73013c8ca4bac2f731159d4b0fe
```

以下是在連接埠 443 上啟用交互 TLS 驗證模式的 HTTPS 接聽程式連線失敗的日誌項目範例。

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
 -
"CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Failed:ClientCertUntrusted TID_1c71a68d70587445ad5127ff8b2687d7
```

處理連線日誌檔案

連線日誌檔案會壓縮。如您利用 Amazon S3 主控台開啟檔案，則會解壓縮檔案並顯示資訊。如果您下載檔案，則必須先將其解壓縮才能看到資訊。

如果您的網站上有許多需求，負載平衡器產生的日誌檔可能有好幾 GB 的資料。您可能無法逐行處理這麼龐大的資料。因此，您可能需要使用提供平行處理解決方案的分析工具。例如，您可以使用下列分析工具來分析和處理連線日誌：

- Amazon Athena 是一種互動式查詢服務，可讓您使用標準 SQL 輕鬆分析 Amazon S3 中的資料。
- Loggly
- Splunk
- Sumo Logic

啟用 Application Load Balancer 的連線日誌

當您為負載平衡器啟用連線日誌時，您必須指定負載平衡器將存放日誌的 S3 儲存貯體名稱。儲存貯體必須具有儲存貯體政策，能授予 Elastic Load Balancing 寫入儲存貯體的許可。

任務

- 步驟 1：建立 S3 儲存貯體
- 步驟 2：連接政策到您的 S3 儲存貯體
- 步驟 3：設定連線日誌
- 步驟 4：確認儲存貯體許可
- 疑難排解

步驟 1：建立 S3 儲存貯體

啟用連線日誌時，您必須為連線日誌指定 S3 儲存貯體。您可以使用現有的儲存貯體，或特別為連線日誌建立儲存貯體。儲存貯體必須符合下列需求。

要求

- 儲存貯體與負載平衡器必須位於相同的 Region (區域)。儲存貯體和負載平衡器可以由不同的帳戶擁有。
- Amazon S3 受管金鑰 (SSE-S3) 是唯一支援的伺服器端加密選項。如需詳細資訊，請參閱 Amazon S3 受管加密金鑰 (SSE-S3)。

使用 Amazon S3 主控台建立 S3 儲存貯體

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。

2. 選擇建立儲存貯體。
3. 在 Create bucket (建立儲存貯體) 頁面上，執行下列操作：
   a. 針對 Bucket name (儲存貯體名稱)，輸入儲存貯體的名稱。該名稱在 Amazon S3 中所有現有的儲存貯體名稱之間，不得重複。在某些區域，可能會對儲存貯體的名稱進行其他限制。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的儲存貯體限制。
   b. 針對 AWS 區域，選取您建立負載平衡器時所在的區域。
   c. 對於預設加密，選擇 Amazon S3 受管金鑰 (SSE-S3)。
   d. 選擇建立儲存貯體。

步驟 2：連接政策到您的 S3 儲存貯體

您的 S3 儲存貯體必須有儲存貯體政策，授予 Elastic Load Balancing 將連線日誌寫入儲存貯體的許可。儲存貯體政策是以存取政策語言所編寫的 JSON 陳述式集合，可定義儲存貯體的存取許可。每個陳述式包含單一許可的相關資訊，且包含一系列的元素。

如果您使用的是已連接政策的現有儲存貯體，您可以將 Elastic Load Balancing 連線日誌的 陳述式新增至政策。如果您這樣做，我們建議您評估產生的一組許可，以確保它們適用於需要存取 儲存貯體以進行連線日誌的使用者。

儲存貯體政策

此政策會將許可授予指定的日誌交付服務。

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
}
```

針對 Resource，使用範例政策中顯示的格式，輸入存取日誌位置的 ARN。一律在 S3 儲存貯體 ARN 的資源路徑中包含具有負載平衡器的帳戶 ID。這可確保只有來自指定帳戶的負載平衡器才能將存取日誌寫入 S3 儲存貯體。

您指定的 ARN 取決於您是否計劃在步驟 3 中啟用存取日誌時包含字首。

字首為 的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket，字首為 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

AWS GovCloud (US) – 下列範例使用 的 ARN 語法 AWS GovCloud (US) Regions。

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

沒有字首的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket。S3 儲存貯體 ARN 中沒有字首部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

AWS GovCloud (US) – 下列範例使用 的 ARN 語法 AWS GovCloud (US) Regions。

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

舊版儲存貯體政策

先前，對於 2022 年 8 月之前可用的區域，我們需要政策，將許可授予該區域特定的 Elastic Load Balancing 帳戶。仍然支援此舊版政策，但建議您將其取代為上述較新的政策。如果您偏好繼續使用此處未顯示的舊版政策，您可以這麼做。

以下是舊版政策中要在 Principal 中指定的 Elastic Load Balancing 帳戶的 IDs，以供參考。請注意，不在此清單中的區域不支援舊版政策。

- 美國東部 (維吉尼亞北部) – 127311923021
- 美國東部 (俄亥俄) – 033677994240
- 美國西部 (加利佛尼亞北部) – 027434742980

- 美國西部 (奧勒岡) – 797873946194
- 非洲 (開普敦) – 098369216593
- 亞太區域 (香港) – 754344448648
- 亞太區域 (雅加達) – 589379963580
- 亞太區域 (孟買) – 718504428378
- 亞太區域 (大阪) – 383597477331
- 亞太區域 (首爾) – 600734575887
- 亞太區域 (新加坡) – 114774131450
- 亞太區域 (雪梨) – 783225319266
- 亞太區域 (東京) – 582318560864
- 加拿大 (中部) – 985666609251
- 歐洲 (法蘭克福) – 054676820928
- 歐洲 (愛爾蘭) – 156460612806
- 歐洲 (倫敦) – 652711504416
- 歐洲 (米蘭) – 635631232127
- 歐洲 (巴黎) – 009996457667
- 歐洲 (斯德哥爾摩) – 897822967062
- 中東 (巴林) – 076674570225
- 南美洲 (聖保羅) – 507241528517
- AWS GovCloud （美國東部） – 190560391635
- AWS GovCloud （美國西部） – 048591011584

Outpost 區域

以下政策會將許可授予指定的日誌交付服務。將此政策用於 Outpost 區域中的負載平衡器。

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logdelivery.elb.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
```

```
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

針對 Resource，輸入存取日誌位置的 ARN。一律在 S3 儲存貯體 ARN 的資源路徑中包含具有負載平衡器的帳戶 ID。這可確保只有來自指定帳戶的負載平衡器才能將存取日誌寫入 S3 儲存貯體。

您指定的 ARN 取決於您是否計劃在步驟 3 中啟用存取日誌時包含字首。

字首為 的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket，字首為 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

沒有字首的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket。S3 儲存貯體 ARN 中沒有字首部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

安全最佳實務

若要增強安全性，請使用精確的 S3 儲存貯ARNs。

- 使用完整的資源路徑，而不只是 S3 儲存貯體 ARN。
- 包含 S3 儲存貯體 ARN 的帳戶 ID 部分。
- 請勿在 S3 儲存貯體 ARN 的帳戶 ID 部分中使用萬用字元 (*)。

建立儲存貯體政策後，請使用 Amazon S3 介面，例如 Amazon S3 主控台或 AWS CLI 命令，將儲存貯體政策連接至 S3 儲存貯體。

Console

將儲存貯體政策連接至 S3 儲存貯體

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。

2. 選取儲存貯體的名稱，開啟其詳細資訊頁面。
3. 選擇 Permissions (許可)，然後選擇 Bucket policy (儲存貯體政策)、Edit (編輯)。
4. 更新儲存貯體政策，授予所需許可。
5. 選擇儲存變更。

AWS CLI

將儲存貯體政策連接至 S3 儲存貯體

使用 put-bucket-policy 命令。在此範例中，儲存貯體政策已儲存至指定的 .json 檔案。

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://access-log-policy.json
```

步驟 3：設定連線日誌

使用下列程序來設定連線日誌，以擷取日誌檔案並將其交付至 S3 儲存貯體。

要求

儲存貯體必須符合步驟 1 中所述的要求，且您必須按照步驟 2 所述連接儲存貯體政策。如果指定字首，則其不得包含字串 "AWSLogs"。

管理連線日誌的 S3 儲存貯體

刪除您為連線日誌設定的儲存貯體之前，請務必停用連線日誌。否則，如果有新儲存貯體具有相同的名稱和所需的儲存貯體政策，但在非您擁有的 AWS 帳戶 中建立，Elastic Load Balancing 可以將負載平衡器的連線日誌寫入此新儲存貯體。

Console

啟用連線日誌

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取您負載平衡器的名稱來開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。

5. 針對監控，開啟連線日誌。
6. 針對 S3 URI，請輸入日誌檔案的 S3 URI。指定的 URI 取決於您是否使用字首。
   - 帶有字首的 URI：`s3://bucket-name/prefix`
   - 不帶字首的 URI：`s3://bucket-name`
7. 選擇儲存變更。

AWS CLI

啟用連線日誌

使用 modify-load-balancer-attributes 命令搭配相關屬性。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes \
        Key=connection_logs.s3.enabled,Value=true \
        Key=connection_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \
        Key=connection_logs.s3.prefix,Value=logging-prefix
```

CloudFormation

啟用連線日誌

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含相關屬性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "connection_logs.s3.enabled"
```

```
        Value: "true"
      - Key: "connection_logs.s3.bucket"
        Value: "amzn-s3-demo-logging-bucket"
      - Key: "connection_logs.s3.prefix"
        Value: "logging-prefix"
```

步驟 4：確認儲存貯體許可

為您的負載平衡器啟用連線日誌後，Elastic Load Balancing 會驗證 S3 儲存貯體並建立測試檔案，以確保儲存貯體政策指定必要的許可。您可以使用 Amazon S3 主控台來確認是否已建立測試檔案。測試檔案不是實際的連線日誌檔案；它不包含範例記錄。

驗證 Elastic Load Balancing 已在 S3 儲存貯體中建立測試檔案

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選取您為連線日誌指定的儲存貯體名稱。
3. 導覽到測試檔案，`ELBConnectionLogTestFile`。位置取決於您是否使用字首。
   - 字首為 *amzn-s3-demo-logging-bucket*/*prefix*/AWSLogs/*123456789012*/ELBConnectionLogTestFile 的位置
   - 沒有字首的位置：*amzn-s3-demo-logging-bucket*/AWSLogs/*123456789012*/ELBConnectionLogTestFile

疑難排解

如果您收到存取遭拒錯誤，則以下是可能的原因：

- 儲存貯體政策不會授予 Elastic Load Balancing 將連線日誌寫入儲存貯體的許可。確認您正在使用適合該區域的正確儲存貯體政策。確認資源 ARN 使用您在啟用連線日誌時指定的相同儲存貯體名稱。如果您在啟用連線日誌時未指定字首，請確認資源 ARN 不包含字首。
- 儲存貯體使用不支援的伺服器端加密選項。儲存貯體必須使用 Amazon S3 受管金鑰 (SSE-S3)。

停用 Application Load Balancer 的連線日誌

您可以隨時停用負載平衡器的連線日誌。停用連線日誌後，您的連線日誌會保留在您的 S3 儲存貯體中，直到您將其刪除為止。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立、設定和使用儲存貯體。

Console

停用連線日誌

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取您負載平衡器的名稱來開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 針對監控，請關閉連線日誌。
6. 選擇儲存變更。

AWS CLI

停用連線日誌

使用 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes Key=connection_logs.s3.enabled,Value=false
```

運作狀態檢查日誌

Elastic Load Balancing 提供運作狀態檢查日誌，可擷取已註冊目標運作狀態檢查狀態的詳細資訊，包括運作狀態檢查失敗時的失敗原因。EC2 執行個體、IP 地址和 Lambda 函數目標支援運作狀態檢查日誌。每個日誌項目都包含運作狀態檢查請求類型或連線、時間戳記、目標地址、目標群組 ID、運作狀態和原因碼等資訊。您可以使用這些運作狀態檢查日誌來分析目標運作狀態模式、監控運作狀態轉換，以及疑難排解問題。

運作狀態檢查日誌是選用功能，預設為停用。啟用負載平衡器的運作狀態檢查日誌後，Elastic Load Balancing 會擷取日誌，並將其儲存為壓縮檔案，存放在您指定的 Amazon S3 儲存貯體中。您可以隨時停用運作狀態檢查日誌。

您將需支付 Amazon S3 的儲存費用，但 Elastic Load Balancing 將日誌檔傳送到 Amazon S3 所使用的頻寬不需要付費。如需有關儲存費用的詳細資訊，請參閱 Amazon S3 定價。

目錄

- 運作狀態檢查日誌檔案
- 運作狀態檢查日誌項目
- 範例日誌項目
- 設定日誌交付通知
- 處理運作狀態檢查日誌檔案
- 啟用 Application Load Balancer 的運作狀態檢查日誌
- 停用 Application Load Balancer 的運作狀態檢查日誌

運作狀態檢查日誌檔案

Elastic Load Balancing 每 5 分鐘發佈每個負載平衡器節點的日誌檔。當大量目標連接到負載平衡器或設定小型運作狀態檢查間隔時（例如，每 5 秒），負載平衡器可以交付多個日誌。

運作狀態檢查日誌的檔案名稱使用以下格式：

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/
health_check_log_aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-
time_ip-address_random-string.log.gz
```

bucket

S3 儲存貯體的名稱。

prefix

(選用) 儲存貯體的字首 (邏輯階層)。您指定的字首不得包含字串 AWSLogs。如需詳細資訊，請參閱使用字首組織物件。

AWSLogs

我們在您指定的儲存貯體名稱和可選字首之後，增加了以 AWSLogs 開頭的檔案名稱部分。

aws-account-id

擁有者的帳戶 AWS ID。

region

負載平衡器和 S3 儲存貯體的區域。

yyyy/mm/dd

傳遞日誌的日期。

load-balancer-id

負載平衡器的資源 ID。如果資源 ID 包含任何斜線 (/)，斜線會換成句點 (.)。

end-time

記錄間隔結束的日期和時間。例如，結束時間 20140215T2340Z 的日子檔案包含在 23:35 和 23:40 (UTC 或 Zulu 時間) 之間發出的請求項目。

ip-address

處理請求之負載平衡器節點的 IP 地址。對於內部負載平衡器，這是私有 IP 地址。

random-string

系統產生的隨機字串。

以下是含字首的日誌檔案名稱範例：

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/
elasticloadbalancing/us-east-2/2022/05/01/
health_check_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

以下是不含字首的日誌檔案名稱範例：

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/health_check_log_123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

日誌檔案可存放於儲存貯體任意長時間，但您也可以定義 Amazon S3 生命週期規則，自動封存或刪除日誌檔案。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的物件生命週期管理。

運作狀態檢查日誌項目

Elastic Load Balancing 會記錄目標運作狀態檢查結果，包括該負載平衡器所有已註冊目標的失敗原因。每個日誌項目都包含對已註冊目標所做的單一運作狀態檢查結果的詳細資訊。

目錄

- 語法
- 錯誤原因代碼

語法

下表依序說明運作狀態檢查日誌項目的欄位。所有欄位以空格分隔。當我們新增欄位時，會將它新增至日誌項目的結尾。當我們準備發佈新欄位時，您可能會在欄位發佈之前看到額外的結尾 "-"。請確定您將日誌剖析設定為在上次記錄欄位後停止，並在我們發佈新欄位後更新日誌剖析。

欄位 （位置）	Description
類型 (1)	運作狀態檢查請求或連線的類型。可能的值如下所示 (忽略任何其他值)： • `http` -- HTTP • `https` -- HTTP over TLS • h2 -- HTTP/2 over TLS • `grpc` -- gRPC • `lambda` -- Lambda 函數
時間 (2)	在目標上啟動運作狀態檢查時的時間戳記，格式為 ISO 8601。
延遲 (3)	完成目前運作狀態檢查所經過的總時間#（以秒為單位）。
target_addr (4)	目標的 IP 地址和連接埠，格式為 IP：Port。如果目標是 Lambda 函數，則為 Lambda 的 ARN。
target_group_id (5)	與目標相關聯的目標群組名稱。
狀態 (6)	運作狀態檢查的狀態。如果運作狀態檢查成功，則此值為PASS#。在運作狀態檢查失敗時，值為FAIL
status_code (7)	從運作狀態檢查請求的目標收到的回應碼。
reason_code (8)	如果運作狀態檢查失敗，則失敗的原因。請參閱 錯誤原因代碼

錯誤原因代碼

如果目標運作狀態檢查失敗，負載平衡器會在運作狀態檢查日誌中記錄下列其中一個原因代碼。

Code	Description
RequestTimedOut	等待回應時運作狀態檢查請求逾時
ConnectionTimedOut	運作狀態檢查失敗，因為 TCP 連線嘗試逾時
ConnectionReset	由於連線重設，運作狀態檢查失敗
ResponseCodeMismatch	目標回應運作狀態檢查請求的 HTTP 狀態碼不符合設定的狀態碼
ResponseStringMismatch	目標傳回的回應內文不包含目標群組運作狀態檢查組態中設定的字串
InternalError	內部負載平衡器錯誤
TargetError	目標傳回 5xx 錯誤碼以回應運作狀態檢查請求
GRPCStatusHeaderEmpty	GRPC 目標回應具有不含值的 grpc-status 標頭
GRPCUnexpectedStatus	GRPC 目標以非預期的 grpc-status 回應

範例日誌項目

以下是運作狀態檢查日誌項目的範例。請注意，範例文字只會出現在多行上，使其更容易閱讀。

以下是成功運作狀態檢查的範例日誌項目。

```
http 2025-10-31T12:44:59.875678Z 0.019584011 172.31.20.97:80 HCLogsTestIPs PASS 200 -
```

以下是失敗運作狀態檢查的範例日誌項目。

```
http 2025-10-31T12:44:58.901409Z 1.121980746 172.31.31.9:80 HCLogsTestIPs FAIL 502
 TargetError
```

設定日誌交付通知

若要在 Elastic Load Balancing 將日誌交付至 S3 儲存貯體時接收通知，請使用 Amazon S3 事件通知。Elastic Load Balancing 使用 PutObject、CreateMultipartUpload 和 POST 物件將日誌交付至 Amazon S3。為了確保您收到所有日誌交付通知，請在組態中包含所有這些物件建立事件。

如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 Amazon S3 事件通知。

處理運作狀態檢查日誌檔案

運作狀態檢查日誌檔案會壓縮。如果您下載檔案，則必須先將其解壓縮才能看到資訊。

如果您的網站上有許多需求，負載平衡器產生的日誌檔可能有好幾 GB 的資料。您可能無法逐行處理這麼龐大的資料。因此，您可能需要使用提供平行處理解決方案的分析工具。例如，您可以使用下列分析工具來分析和處理運作狀態檢查日誌：

- Amazon Athena 是一種互動式查詢服務，可讓您使用標準 SQL 輕鬆分析 Amazon S3 中的資料。
- Loggly
- Splunk
- Sumo Logic

啟用 Application Load Balancer 的運作狀態檢查日誌

當您啟用負載平衡器的運作狀態檢查日誌時，您必須指定負載平衡器將存放日誌的 S3 儲存貯體名稱。儲存貯體必須具有儲存貯體政策，能授予 Elastic Load Balancing 寫入儲存貯體的許可。

任務

- 步驟 1：建立 S3 儲存貯體
- 步驟 2：連接政策到您的 S3 儲存貯體
- 步驟 3：設定運作狀態檢查日誌
- 步驟 4：確認儲存貯體許可
- 疑難排解

步驟 1：建立 S3 儲存貯體

啟用運作狀態檢查日誌時，您必須為運作狀態檢查日誌指定 S3 儲存貯體。您可以使用現有的儲存貯體，或特別為運作狀態檢查日誌建立儲存貯體。儲存貯體必須符合下列需求。

要求

- 儲存貯體與負載平衡器必須位於相同的 Region (區域)。儲存貯體和負載平衡器可以由不同的帳戶擁有。
- Amazon S3 受管金鑰 (SSE-S3) 是唯一支援的伺服器端加密選項。如需詳細資訊，請參閱 Amazon S3 受管加密金鑰 (SSE-S3)。

使用 Amazon S3 主控台建立 S3 儲存貯體

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選擇建立儲存貯體。
3. 在 Create bucket (建立儲存貯體) 頁面上，執行下列操作：
   a. 針對 Bucket name (儲存貯體名稱)，輸入儲存貯體的名稱。該名稱在 Amazon S3 中所有現有的儲存貯體名稱之間，不得重複。在某些區域，可能會對儲存貯體的名稱進行其他限制。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的儲存貯體限制。
   b. 針對 AWS 區域，選取您建立負載平衡器時所在的區域。
   c. 對於預設加密，選擇 Amazon S3 受管金鑰 (SSE-S3)。
   d. 選擇建立儲存貯體。

步驟 2：連接政策到您的 S3 儲存貯體

您的 S3 儲存貯體必須有儲存貯體政策，授予 Elastic Load Balancing 將運作狀態檢查日誌寫入儲存貯體的許可。儲存貯體政策是以存取政策語言所編寫的 JSON 陳述式集合，可定義儲存貯體的存取許可。每個陳述式包含單一許可的相關資訊，且包含一系列的元素。

如果您使用的是已連接政策的現有儲存貯體，您可以將 Elastic Load Balancing 運作狀態檢查日誌的 陳述式新增至政策。如果您這樣做，我們建議您評估產生的一組許可，以確保它們適用於需要存取儲存貯體以進行運作狀態檢查日誌的使用者。

儲存貯體政策

此政策會將許可授予指定的日誌交付服務。

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
}
```

針對 `Resource`，使用範例政策中顯示的格式，輸入存取日誌位置的 ARN。一律在 S3 儲存貯體 ARN 的資源路徑中包含具有負載平衡器的帳戶 ID。這可確保只有來自指定帳戶的負載平衡器才能將存取日誌寫入 S3 儲存貯體。

您指定的 ARN 取決於您是否計劃在步驟 3 中啟用存取日誌時包含字首。

字首為 的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket，字首為 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

AWS GovCloud (US) – 下列範例使用 的 ARN 語法 AWS GovCloud (US) Regions。

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

沒有字首的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket。S3 儲存貯體 ARN 中沒有字首部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

AWS GovCloud (US) – 下列範例使用 的 ARN 語法 AWS GovCloud (US) Regions。

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

舊版儲存貯體政策

先前，對於 2022 年 8 月之前可用的區域，我們需要一個政策，將許可授予該區域特定的 Elastic Load Balancing 帳戶。仍然支援此舊版政策，但建議您將其取代為上述較新的政策。如果您偏好繼續使用此處未顯示的舊版政策，您可以這麼做。

以下是舊版政策中要在 `Principal` 中指定的 Elastic Load Balancing 帳戶的 IDs，以供參考。請注意，不在此清單中的區域不支援舊版政策。

- 美國東部 (維吉尼亞北部) – 127311923021
- 美國東部 (俄亥俄) – 033677994240
- 美國西部 (加利佛尼亞北部) – 027434742980
- 美國西部 (奧勒岡) – 797873946194
- 非洲 (開普敦) – 098369216593
- 亞太區域 (香港) – 754344448648
- 亞太區域 (雅加達) – 589379963580
- 亞太區域 (孟買) – 718504428378
- 亞太區域 (大阪) – 383597477331
- 亞太區域 (首爾) – 600734575887
- 亞太區域 (新加坡) – 114774131450
- 亞太區域 (雪梨) – 783225319266
- 亞太區域 (東京) – 582318560864
- 加拿大 (中部) – 985666609251
- 歐洲 (法蘭克福) – 054676820928
- 歐洲 (愛爾蘭) – 156460612806
- 歐洲 (倫敦) – 652711504416
- 歐洲 (米蘭) – 635631232127
- 歐洲 (巴黎) – 009996457667
- 歐洲 (斯德哥爾摩) – 897822967062
- 中東 (巴林) – 076674570225
- 南美洲 (聖保羅) – 507241528517
- AWS GovCloud （美國東部） – 190560391635
- AWS GovCloud （美國西部） – 048591011584

Outpost 區域

以下政策會將許可授予指定的日誌交付服務。將此政策用於 Outpost 區域中的負載平衡器。

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logdelivery.elb.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

針對 Resource，輸入存取日誌位置的 ARN。一律在 S3 儲存貯體 ARN 的資源路徑中包含具有負載平衡器的帳戶 ID。這可確保只有來自指定帳戶的負載平衡器才能將存取日誌寫入 S3 儲存貯體。

您指定的 ARN 取決於您是否計劃在步驟 3 中啟用存取日誌時包含字首。

字首為 的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket，字首為 logging-prefix。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

沒有字首的範例 S3 儲存貯體 ARN

S3 儲存貯體名稱為 amzn-s3-demo-logging-bucket。S3 儲存貯體 ARN 中沒有字首部分。

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

安全最佳實務

若要增強安全性，請使用精確的 S3 儲存貯ARNs。

- 使用完整資源路徑，而不只是 S3 儲存貯體 ARN。
- 包含 S3 儲存貯體 ARN 的帳戶 ID 部分。

- 請勿在 S3 儲存貯體 ARN 的帳戶 ID 部分中使用萬用字元 (*)。

建立儲存貯體政策後，請使用 Amazon S3 介面，例如 Amazon S3 主控台或 AWS CLI 命令，將儲存貯體政策連接至 S3 儲存貯體。

Console

將儲存貯體政策連接至 S3 儲存貯體

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選取儲存貯體的名稱，開啟其詳細資訊頁面。
3. 選擇 Permissions (許可)，然後選擇 Bucket policy (儲存貯體政策)、Edit (編輯)。
4. 更新儲存貯體政策，授予所需許可。
5. 選擇儲存變更。

AWS CLI

將儲存貯體政策連接至 S3 儲存貯體

使用 put-bucket-policy 命令。在此範例中，儲存貯體政策已儲存至指定的 .json 檔案。

```
aws s3api put-bucket-policy \
    --bucket amzn-s3-demo-bucket \
    --policy file://access-log-policy.json
```

步驟 3：設定運作狀態檢查日誌

使用下列程序來設定運作狀態檢查日誌，以擷取日誌檔案並將其交付至 S3 儲存貯體。

要求

儲存貯體必須符合步驟 1 中所述的要求，且您必須按照步驟 2 所述連接儲存貯體政策。如果指定字首，則其不得包含字串 "AWSLogs"。

管理運作狀態檢查日誌的 S3 儲存貯體

刪除您為運作狀態檢查日誌設定的儲存貯體之前，請務必停用運作狀態檢查日誌。否則，如果有具有相同名稱的新儲存貯體和所需的儲存貯體政策，但在非您擁有的 AWS 帳戶 中建立，Elastic Load Balancing 可以將負載平衡器的運作狀態檢查日誌寫入此新儲存貯體。

Console

啟用運作狀態檢查日誌

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取您負載平衡器的名稱來開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 針對監控，開啟運作狀態檢查日誌。
6. 針對 S3 URI，請輸入日誌檔案的 S3 URI。指定的 URI 取決於您是否使用字首。
   - 帶有字首的 URI：`s3://bucket-name/prefix`
   - 不帶字首的 URI：`s3://bucket-name`
7. 選擇儲存變更。

AWS CLI

啟用運作狀態檢查日誌

使用 modify-load-balancer-attributes 命令搭配相關屬性。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes \
        Key=health_check_logs.s3.enabled,Value=true \
        Key=health_check_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \
        Key=health_check_logs.s3.prefix,Value=logging-prefix
```

CloudFormation

啟用運作狀態檢查日誌

更新 AWS::ElasticLoadBalancingV2::LoadBalancer 資源以包含相關屬性。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
```

```
    Scheme: internal
    Subnets:
      - !Ref subnet-AZ1
      - !Ref subnet-AZ2
    SecurityGroups:
      - !Ref mySecurityGroup
    LoadBalancerAttributes:
      - Key: "health_check_logs.s3.enabled"
        Value: "true"
      - Key: "health_check_logs.s3.bucket"
        Value: "amzn-s3-demo-logging-bucket"
      - Key: "health_check_logs.s3.prefix"
        Value: "logging-prefix"
```

步驟 4：確認儲存貯體許可

為您的負載平衡器啟用運作狀態檢查日誌後，Elastic Load Balancing 會驗證 S3 儲存貯體並建立測試檔案，以確保儲存貯體政策指定必要的許可。您可以使用 Amazon S3 主控台來確認是否已建立測試檔案。測試檔案不是實際的運作狀態檢查日誌檔案；它不包含範例記錄。

驗證 Elastic Load Balancing 已在 S3 儲存貯體中建立測試檔案

1. 開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
2. 選取您為運作狀態檢查日誌指定的儲存貯體名稱。
3. 導覽到測試檔案，ELBHealthCheckLogTestFile。位置取決於您是否使用字首。

   - 字首為 *amzn-s3-demo-logging-bucket*/*prefix*/AWSLogs/*123456789012*/ELBHealthCheckLogTestFile 的位置
   - 沒有字首的位置：*amzn-s3-demo-logging-bucket*/AWSLogs/*123456789012*/ELBHealthCheckLogTestFile

疑難排解

如果您收到存取遭拒錯誤，則以下是可能的原因：

- 儲存貯體政策不會授予 Elastic Load Balancing 將運作狀態檢查日誌寫入儲存貯體的許可。確認您正在使用適合該區域的正確儲存貯體政策。驗證資源 ARN 是否使用您在啟用運作狀態檢查日誌時指定的相同儲存貯體名稱。如果您在啟用運作狀態檢查日誌時未指定字首，請確認資源 ARN 不包含字首。

- 儲存貯體使用不支援的伺服器端加密選項。儲存貯體必須使用 Amazon S3 受管金鑰 (SSE-S3)。

停用 Application Load Balancer 的運作狀態檢查日誌

您可以隨時停用負載平衡器的運作狀態檢查日誌。停用運作狀態檢查日誌後，您的運作狀態檢查日誌會保留在您的 S3 儲存貯體中，直到您將其刪除為止。如需詳細資訊，請參閱《Amazon S3 使用者指南》中的建立、設定和使用儲存貯體。

Console

停用運作狀態檢查日誌

1. 前往 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 在導覽窗格中，選擇 Load Balancers (負載平衡器)。
3. 選取您負載平衡器的名稱來開啟其詳細資訊頁面。
4. 在屬性索引標籤中，選擇編輯。
5. 針對監控，請關閉運作狀態檢查日誌。
6. 選擇儲存變更。

AWS CLI

停用運作狀態檢查日誌

使用 modify-load-balancer-attributes 命令。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes Key=health_check_logs.s3.enabled,Value=false
```

Application Load Balancer 上的請求追蹤

當負載平衡器收到用戶端的請求時，在將請求傳送到目標之前，它會新增或更新 X-Amzn-Trace-Id 標頭。負載平衡器和目標之間的任何服務或應用程式也可以新增或更新此標頭。

您可以使用請求追蹤來追蹤從用戶端到目標或其他服務的 HTTP 請求。如果您啟用存取日誌，則會記錄 X-Amzn-Trace-Id 標頭的內容。如需詳細資訊，請參閱Application Load Balancer 的存取日誌。

語法

X-Amzn-Trace-Id 標頭包含如下格式的欄位：

```
Field=version-time-id
```

欄位

欄位的名稱。支援的值為 Root 和 Self。

應用程式可以新增任意欄位供自己使用。負載平衡器會保留這些欄位，但不使用。

version

版本號碼。此值為 1。

time

epoch 時間 (以秒為單位)。此值長度為 8 個十六進位數字。

id

追蹤識別符。此值為 24 個十六進位數字。

範例

如果傳入請求上沒有 X-Amzn-Trace-Id 標頭，負載平衡器會產生含有 Root 欄位的標頭，然後轉送請求。例如：

```
X-Amzn-Trace-Id: Root=1-67891233-abcdef012345678912345678
```

如果 X-Amzn-Trace-Id 標頭存在且有 Root 欄位，負載平衡器會插入 Self 欄位，然後轉送請求。例如：

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-
abcdef012345678912345678
```

如果應用程式新增含有 Root 欄位和自訂欄位的標頭，負載平衡器會保留這兩個欄位、插入 Self 欄位，然後轉送請求：

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-
abcdef012345678912345678;CalledFrom=app
```

如果 X-Amzn-Trace-Id 標頭存在且有 `Self` 欄位，負載平衡器會更新 `Self` 欄位的值。

限制

- 負載平衡器會在收到傳入請求時，而不是在收到回應時更新標頭。
- 如果 HTTP 標頭大於 7 KB，負載平衡器會使用 `Root` 欄位重寫 X-Amzn-Trace-Id 標頭。
- 使用 WebSocket 時，只有等到升級請求成功時才能追蹤。

為 Application Load Balancer 進行疑難排解

以下資訊有助於您就 Application Load Balancer 的問題進行疑難排解。

問題

- 已註冊目標處於非服務中狀態
- 用戶端無法連接到面向網際網路的負載平衡器
- 負載平衡器不會收到傳送至自訂域的請求
- 傳送至負載平衡器的 HTTPS 要求會傳回 "NET::ERR_CERT_COMMON_NAME_INVALID"
- 負載平衡器顯示處理時間延長
- 負載平衡器會傳送 000 的回應代碼
- 負載平衡器產生 HTTP 錯誤
- 目標產生了 HTTP 錯誤
- AWS Certificate Manager 憑證無法使用
- 不支援多行標頭
- 使用資源映射對運作狀態不佳的目標進行故障診斷
- 針對目標最佳化工具進行故障診斷

已註冊目標處於非服務中狀態

如果目標進入 `InService` 狀態所花的時間超過預期，表示該目標可能未通過運作狀態檢查。您的目標將處於非服務中狀態，除非通過一次運作狀態檢查。如需詳細資訊，請參閱Application Load Balancer 目標群組的運作狀態檢查。

確認您的執行個體是否未通過運作狀態檢查，然後檢查以下問題：

安全群組不允許流量

> 與執行個體相關聯的安全群組，必須允許來自負載平衡器使用運作狀態檢查連接埠和運作狀態檢查通訊協定傳來的流量。您可以將規則新增到執行個體安全群組，以允許來自負載平衡器安全群組的所有流量。此外，負載平衡器的安全群組必須允許對執行個體的流量。

網路存取控制清單 (ACL) 不允許流量

與執行個體的子網路相關聯的網路 ACL 必須允許透過運作狀態檢查連接埠傳送對內流量，以及透過暫時性連接埠 (1024-65535) 傳送對外流量。與負載平衡器節點的子網路相關聯的網路 ACL 必須允許透過暫時性連接埠傳送對內流量，以及透過運作狀態檢查連接埠和暫時性連接埠傳送對外流量。

ping 路徑不存在

針對運作狀態檢查建立目標頁面，並指定其路徑做為 ping 路徑。

連線逾時

首先，驗證您可以使用目標的私有 IP 地址和運作狀態檢查通訊協定，從網路內直接連接到目標。如果無法連接，請檢查是否過度利用該執行個體，如果它太忙碌而無法回應，便將更多目標新增到您的目標群組。如果您可以連接，則在運作狀態檢查逾時期間之前，目標頁面可能都沒有回應。為運作狀態檢查選擇較簡單的目標頁面，或調整運作狀態檢查設定。

目標未傳回成功的回應代碼

在預設情況下，成功代碼是 200，但您可以在設定運作狀態檢查時選擇性地指定額外的成功代碼。確認負載平衡器預期的成功代碼，並且您的應用程式已設定為在成功時傳回這些代碼。

目標回應代碼錯誤或連線至目標時發生錯誤

確認應用程式是否回應負載平衡器的運作狀態檢查請求。某些應用程式需要額外的組態才能回應運作狀態檢查，例如需要有虛擬主機組態才能回應負載平衡器傳送的 HTTP 主機標頭。主機標頭值包含目標的私有 IP 地址，後面接著不使用預設連接埠時的運作狀態檢查連接埠。如果目標使用預設的運作狀態檢查連接埠，主機標頭值只會包含目標的私有 IP 地址。例如，如果您目標的私有 IP 地址為 ，`10.0.0.10`且其運作狀態檢查連接埠為 `8080`，則運作狀態檢查中負載平衡器傳送的 HTTP 主機標頭為 `Host: 10.0.0.10:8080`。如果您目標的私有 IP 地址為 ，`10.0.0.10`且其運作狀態檢查連接埠為 `80`，則運作狀態檢查中負載平衡器傳送的 HTTP 主機標頭為 `Host: 10.0.0.10`。可能需要虛擬主機組態來回應該主機，或是預設組態，才能順利進行應用程式的運作狀態檢查。運作狀態檢查請求具有下列屬性：`User-Agent` 設定為 `ELB-HealthChecker/2.0`，訊息標頭欄位的行結束字元是 CRLF 序列，標頭會在第一個空白行終止，後面接著 CRLF。

用戶端無法連接到面向網際網路的負載平衡器

如果負載平衡器未回應請求，則請檢查下列問題：

您的面向網際網路的負載平衡器已連接到私有子網路

您必須為負載平衡器指定公有子網路。公有子網路具有適用您虛擬私有雲端 (VPC) 對網際網路閘道的路由。

安全群組或網路 ACL 不允許流量

負載平衡器的安全群組和負載平衡器子網路的任何網路 ACL，必須允許來自用戶端的傳入流量和連至接聽程式連接埠上用戶端的傳出流量。

負載平衡器不會收到傳送至自訂域的請求

如果負載平衡器未收到傳送至自訂域的請求，則請檢查下列問題：

自訂域名稱未解析為負載平衡器 IP 地址

- 使用命令列介面確認自訂域名稱解析的目標 IP 地址。
  - Linux、macOS 或 Unix – 您可以在終端內使用 `dig` 命令。例如 `dig example.com`
  - Windows – 您可以在命令提示內使用 `nslookup` 命令。例如 `nslookup example.com`
- 使用命令列介面確認負載平衡器 DNS 名稱解析的目標 IP 地址。
- 比較兩種輸出的結果。IP 地址必須相符。

如果使用 Route 53 託管自訂網域，請參閱《Amazon Route 53 開發人員指南》中的我的網域在網際網路不可用。

傳送至負載平衡器的 HTTPS 要求會傳回 "NET::ERR_CERT_COMMON_NAME_INVALID"

如果 HTTPS 請求從負載平衡器接收 `NET::ERR_CERT_COMMON_NAME_INVALID`，則請檢查下列可能的原因：

- HTTPS 請求中使用的域名稱與在關聯 ACM 憑證之接聽程式中指定的替代名稱不相符。
- 正在使用負載平衡器預設 DNS 名稱。預設 DNS 名稱無法用於提出 HTTPS 請求，因為無法針對 `*.amazonaws.com` 域請求公有憑證。

負載平衡器顯示處理時間延長

負載平衡器會根據組態以不同的方式計算處理時間。

- 如果 AWS WAF 與您的 Application Load Balancer 相關聯，且用戶端傳送 HTTP POST 請求，則傳送 POST 請求資料的時間會反映在負載平衡器存取日誌中的 `request_processing_time` 欄位。HTTP POST 請求預期會發生這種行為。
- 如果 AWS WAF 未與您的 Application Load Balancer 建立關聯，且用戶端傳送 HTTP POST 請求，則傳送 POST 請求資料的時間會反映在負載平衡器存取日誌中的 `target_processing_time` 欄位。HTTP POST 請求預期會發生這種行為。

負載平衡器會傳送 000 的回應代碼

使用 HTTP/2 連線時，如果透過一個連線提供的請求數量超過 10，000，負載平衡器會傳送 GOAWAY 框架並關閉與 TCP FIN 的連線。

負載平衡器產生 HTTP 錯誤

以下 HTTP 錯誤是由負載平衡器產生。負載平衡器會將 HTTP 程式碼傳送到用戶端，將請求儲存到存取日誌，並遞增 `HTTPCode_ELB_4XX_Count` 或 `HTTPCode_ELB_5XX_Count` 指標。

錯誤

- HTTP 400：錯誤的請求
- HTTP 401：未經授權
- HTTP 403：禁止
- HTTP 405：方法不允許
- HTTP 408：請求逾時
- HTTP 413：承載過大
- HTTP 414：URI 過長
- HTTP 460
- HTTP 463
- HTTP 464
- HTTP 500：內部伺服器錯誤

- HTTP 501：未導入
- HTTP 502：無效的閘道
- HTTP 503：服務無法使用
- HTTP 504：閘道逾時
- HTTP 505：不支援的版本
- HTTP 507：儲存不足
- HTTP 561：未經授權
- HTTP 562：JWKS 請求失敗

HTTP 400：錯誤的請求

可能原因：

- 用戶端傳送不符合 HTTP 規格的格式錯誤請求。
- 請求標頭超過每個請求行 16 K、每個單一標頭 16 K，或整個請求標頭 64 K 的限制。
- 用戶端在傳送完整請求內文之前關閉了連線。

HTTP 401：未經授權

您設定了接聽程式規則來驗證使用者，但下列其中一項成立：

- 您已將 `OnUnauthenticatedRequest` 設定為拒絕未經身分驗證的使用者，或是 IdP 拒絕了存取。
- IdP 傳回的宣告大小超過負載平衡器支援的大小上限。
- 用戶端提交了不含主機標頭的 HTTP/1.0 請求，負載平衡器無法產生重新導向 URL。
- 請求的範圍不會傳回 ID 字符。
- 您沒有在用戶端登入逾時到期之前完成登入程序。如需詳細資訊，請參閱 Client login timeout。
- 由於下列其中一個原因，JWT 身分驗證失敗：
  - 請求缺少授權標頭。(JWTHeaderNotPresent)
  - 請求中的字符格式無效。這可能發生在下列情況：
    - 字符格式錯誤或缺少必要部分（標頭、承載或簽章）
    - 標頭缺少「承載」字首
    - 標頭包含不同的身分驗證類型（例如「基本」）

    - 授權標頭存在，但字符遺失
    - 請求中存在多個字符 (JWTRequestFormatInvalid)
- 字符簽章驗證失敗。這可能發生在下列情況：
    - 簽章不相符
    - 公有金鑰無效或無法轉換為解碼金鑰
    - 公有金鑰大小不是 2K
    - 權杖使用不支援的演算法簽署
    - 字符中的 KID 不存在於 JWKS 端點 (JWTSignatureValidationFailed)
- JWT 缺少驗證所需的宣告。(JWTClaimNotPresent)
- JWT 中宣告值的格式不符合指定的組態格式。(JWTClaimFormatInvalid)

HTTP 403：禁止

您已設定 AWS WAF Web 存取控制清單 (Web ACL) 來監控對 Application Load Balancer 的請求，並封鎖請求。

HTTP 405：方法不允許

用戶端使用了 TRACE 方法，該方法不受 Application Load Balancer 支援。

HTTP 408：請求逾時

用戶端在閒置逾時期間過期前不會傳送資料。傳送 TCP 持續作用無法防止此逾時。在每個閒置逾時期間經過前，先傳送至少 1 位元組的資料。視需要提高閒置逾時期間的長度。

HTTP 413：承載過大

可能原因：

- 目標是 Lambda 函數，請求內文超過 1 MB。
- 請求標頭超過每個請求行 16 K、每個單一標頭 16 K，或整個請求標頭 64 K 的限制。

HTTP 414：URI 過長

請求 URL 或查詢字串參數太大。

HTTP 460

負載平衡器收到來自用戶端的請求，但用戶端在閒置逾時期間經過之前關閉了與負載平衡器的連線。

檢查用戶端逾時期間是否大於負載平衡器的閒置逾時期間。確保您的目標在用戶端逾時期間經過之前向用戶端提供回應，或如果用戶端支援的話，增加用戶端逾時期間以符合負載平衡器閒置逾時。

HTTP 463

負載平衡器收到具有過多 IP 地址的 X-Forwarded-For 請求標頭。IP 地址的數量上限為 30。

HTTP 464

負載平衡器收到的傳入請求通訊協定與目標群組通訊協定的版本組態不相容。

可能原因：

- 請求通訊協定是 HTTP/1.1，而目標群組通訊協定版本是 gRPC 或 HTTP/2。
- 請求通訊協定是 gRPC，而目標群組通訊協定版本是 HTTP/1.1。
- 請求通訊協定是 HTTP/2，請求不是 POST，而目標群組通訊協定版本是 gRPC。

HTTP 500：內部伺服器錯誤

可能原因：

- 您已設定 AWS WAF Web 存取控制清單 (Web ACL)，但執行 Web ACL 規則時發生錯誤。
- 負載平衡器無法與 IdP 字符端點或 IdP 使用者資訊端點通訊。
  - 確認 IdP 的 DNS 是否可公開解析。
  - 驗證您的負載平衡器的安全群組和您的 VPC 的網路 ACL 允許對這些端點的傳出存取。
  - 驗證您的 VPC 具有網際網路存取。如果您有面對內部的負載平衡器，請使用 NAT 閘道來啟用網際網路存取。
- 從 IdP 收到的使用者宣告大小大於 11KB。
- IdP 字符端點或 IdP 使用者資訊端點的回應時間超過 5 秒。
- 負載平衡器無法與 JWKS 端點通訊，或 JWKS 端點未在 5 秒內回應。
- JWKS 端點傳回的回應大小超過 150KB，或 JWKS 端點傳回的金鑰數目超過 10
- 目標群組已啟用目標最佳化工具，且代理程式遇到未預期的錯誤。請參閱 the section called “針對目標最佳化工具進行故障診斷”。

HTTP 501：未導入

可能原因：

- 負載平衡器收到的 Transfer-Encoding 標頭具有不支援的值。Transfer-Encoding 的支援值為 `chunked` 和 `identity`。或者，您可以使用 Content-Encoding 標頭。
- Websocket 請求已路由至已啟用目標最佳化工具的目標群組。

HTTP 502：無效的閘道

可能原因：

- 在嘗試建立連線之前，負載平衡器從目標接收 TCP RST。
- 在嘗試建立連線之前，負載平衡器從目標收到意外的回應，例如「ICMP 目的地無法連線 (主機無法連線)」。檢查是否允許從負載平衡器子網路對目標連接埠上之目標的流量。
- 當負載平衡器有對目標的未完成請求時，目標關閉了具有 TCP RST 或 TCP FIN 的連線。檢查目標的持續作用持續期間是否短於負載平衡器的閒置逾時值。
- 目標回應的格式錯誤或包含無效的 HTTP 標頭。
- 目標回應標頭超過整個回應標頭 32 K 的限制。
- 由已取消註冊目標處理的請求取消註冊延遲期間已經過。增加延遲期間，使得耗時操作可以完成。
- 目標是 Lambda 函數，回應內文超過 1 MB。
- 目標是一個 Lambda 函數，它沒有在到達設定的逾時之前回應。
- 目標是傳回錯誤的 Lambda 函數，或是受 Lambda 服務限流的函數。
- 負載平衡器在連線至目標時遇到 SSL 交握錯誤。

如需詳細資訊，請參閱 AWS 支援知識中心中的如何疑難排解 Application Load Balancer HTTP 502 錯誤。

HTTP 503：服務無法使用

可能原因：

- 負載平衡器的目標群組沒有已註冊的目標，或所有已註冊的目標都處於 `unused` 狀態。
- 請求已路由至已啟用目標最佳化工具的目標群組，且因為沒有目標已準備好接收請求而遭到拒絕。請參閱 the section called “針對目標最佳化工具進行故障診斷”。

HTTP 504：閘道逾時

可能原因：

- 負載平衡器無法在連線逾時過期 (10 秒) 之前建立對目標的連線。
- 負載平衡器建立了對目標的連線，但目標未在閒置逾時期間經過之前回應。
- 子網路的網路 ACL 不允許從目標到暫時性連接埠 (1024-65535) 上負載平衡器節點的流量。
- 目標傳回的內容長度標頭大於實體主體。負載平衡器等候遺失的位元組時逾時。
- 目標是 Lambda 函數，且 Lambda 服務未在連線逾時期間經過之前回應。
- 負載平衡器在連線至目標時遇到 SSL 交握逾時 (10 秒）。

HTTP 505：不支援的版本

負載平衡器收到非預期的 HTTP 版本請求。例如，負載平衡器建立了 HTTP/1 連線，但收到 HTTP/2 請求。

HTTP 507：儲存不足

重新導向 URL 太長。

HTTP 561：未經授權

您設定了接聽程式規則來驗證使用者，但 IdP 在驗證使用者時傳回錯誤碼。檢查您的存取日誌，以獲取相關的錯誤原因代碼。

HTTP 562：JWKS 請求失敗

負載平衡器無法從 JWKS (JSON Web 金鑰集） 端點接收成功且有效的回應。成功的回應應具有 200-299 範圍內的狀態碼，但改為收到不同的狀態碼。有效的回應不應有下列問題：

- 非 JSON 格式
- 無效的字元
- 無效的 JWKS 格式
- 缺少/無效的必要 JWKS 屬性
- 公有金鑰具有不支援的演算法
- 公有金鑰無法轉換為解碼金鑰

- 公有金鑰大小不是 2K

目標產生了 HTTP 錯誤

負載平衡器將來自目標的有效 HTTP 回應轉送至用戶端，包括 HTTP 錯誤。目標產生的 HTTP 錯誤會記錄在 `HTTPCode_Target_4XX_Count` 和 `HTTPCode_Target_5XX_Count` 指標中。

AWS Certificate Manager 憑證無法使用

決定搭配 Application Load Balancer 使用 HTTPS 接聽程式時， AWS Certificate Manager 會要求您在發出憑證之前驗證網域擁有權。如果在設定期間遺漏此步驟，則憑證會保持在 `Pending Validation` 狀態，且在驗證之前無法使用。

- 如果使用電子郵件驗證，請參閱《AWS Certificate Manager 使用者指南》中的電子郵件驗證。
- 如果使用 DNS 驗證，請參閱《AWS Certificate Manager 使用者指南》中的 DNS 驗證。

不支援多行標頭

Application Load Balancer 不支援多行標頭，包括 `message/http` 媒體類型標頭。如果收到多行標頭，Application Load Balancer 會在將其傳遞至目標之前附加冒號字元 ":"。

使用資源映射對運作狀態不佳的目標進行故障診斷

如果您的 Application Load Balancer 目標未通過運作狀態檢查，您可以使用資源映射來尋找運作狀態不佳的目標，並根據失敗原因代碼採取動作。如需詳細資訊，請參閱檢視 Application Load Balancer 資源映射。

資源映射提供兩種檢視：概觀和運作狀態不佳的目標映射。預設會選取概觀，並顯示所有負載平衡器的資源。選取運作狀態不佳的目標映射檢視只會顯示與 Application Load Balancer 相關聯的每個目標群組中運作狀態不佳的目標。

> **Note**
>
> 您必須啟用顯示資源詳細資訊，才能檢視資源映射中所有適用資源的運作狀態檢查摘要和錯誤訊息。未啟用時，您必須選取每個資源以檢視其詳細資訊。

目標群組欄會顯示每個目標群組運作狀態良好和運作狀態不佳目標的摘要。這有助於判斷所有目標是否未通過運作狀態檢查，或只有特定目標未通過。如果目標群組中的所有目標都未通過運作狀態檢查，請檢查目標群組的組態。選取目標群組名稱，以在新索引標籤中開啟其詳細資訊頁面。

目標欄會顯示每個目標的 TargetID 和目前運作狀態檢查狀態。當目標運作狀態不佳時，會顯示運作狀態檢查失敗原因代碼。當單一目標未通過運作狀態檢查時，請確認目標有足夠的資源，並確認目標上執行的應用程式可用。選取目標 ID，在新標籤中開啟其詳細資訊頁面。

選取匯出可讓您選擇將 Application Load Balancer 資源映射的目前檢視匯出為 PDF。

確認您的執行個體運作狀態檢查失敗，然後根據故障原因程式碼檢查下列問題：

- 運作狀態不佳：HTTP 回應不相符
  - 確認目標上執行的應用程式傳送正確的 HTTP 回應給 Application Load Balancer 的運作狀態檢查請求。
  - 或者，您可以更新 Application Load Balancer 的運作狀態檢查請求，以符合目標上執行之應用程式的回應。
- 運作狀態不佳：請求逾時
  - 確認與您的目標和 Application Load Balancer 相關聯的安全群組和網路存取控制清單 (ACL) 未封鎖連線。
  - 確認目標有足夠的資源可以接受來自 Application Load Balancer 的連線。
  - 驗證在目標上執行的任何應用程式的狀態。
  - Application Load Balancer 的運作狀態檢查回應可以在每個目標的應用程式日誌中檢視。如需詳細資訊，請參閱運作狀態檢查原因代碼。
- 運作狀態不佳：FailedHealthChecks
  - 驗證在目標上執行的任何應用程式的狀態。
  - 確認目標正在接聽運作狀態檢查連接埠上的流量。

> **使用 HTTPS 接聽程式時**
>
> 您可以選擇用於前端連線的安全政策。用於後端連線的安全政策會根據使用的前端安全政策自動選取。如果您的任何接聽程式有：
>
> - FIPS 後量子 TLS 政策 - 後端連線使用 `ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09`
> - FIPS 政策 - 後端連線使用 `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04`
> - 後量子 TLS 政策 - 後端連線使用 `ELBSecurityPolicy-TLS13-1-0-PQ-2025-09`

- TLS 1.3 政策 - 後端連線使用 ELBSecurityPolicy-TLS13-1-0-2021-06
- 所有其他 TLS 政策後端連線都使用 ELBSecurityPolicy-2016-08

如需詳細資訊，請參閱 安全政策。

- 驗證目標是否以安全政策指定的正確格式提供伺服器憑證和金鑰。
- 確認目標支援一或多個相符的密碼，以及 Application Load Balancer 提供的通訊協定來建立 TLS 交握。

針對目標最佳化工具進行故障診斷

如需詳細監控，請參閱目標最佳化工具指標

組態錯誤

- HTTPCode_ELB_502_Count：負載平衡器在嘗試建立連線時收到來自代理程式的 TCP RST。
- HTTPCode_ELB_504_Count：負載平衡器無法在閒置逾時期間之前建立與代理程式的連線。
- HTTPCode_Target_5XX_Count：代理程式在嘗試建立連線時，從目標應用程式收到 TCP RST。（僅適用於目標應用程式本身未產生此錯誤回應時。）

若要修正這些問題，請確保：

- 目標上的安全群組已正確設定。
- 代理程式正在以預期的組態執行。
- 目標應用程式正在代理程式中設定的 TARGET_CONTROL_DESTINATION_ADDRESS 上執行和接聽。

服務無法使用錯誤 (**HTTPCode_ELB_503_Count**)

一致的 HTTP 503 錯誤表示目標不足，無法接收來自 ALB 的請求。TargetControlRequestRejectCount 指標代表這些遭拒的請求。TargetControlWorkQueueLength 指標會降至接近零的值。若要修正此問題，請考慮：

- 增加目標數量
- 將代理程式上的 TARGET_CONTROL_MAX_CONCURRENCY 變數設為較大的值。

運作狀態檢查錯誤

- 如果運作狀態檢查連接埠與 TARGET_CONTROL_DATA_ADDRESS 相同，則來自 ALB 的運作狀態檢查請求會透過代理程式傳送至目標應用程式。如果運作狀態檢查失敗（由於 HTTP 502 或逾時），請參閱組態錯誤一節。

Application Load Balancer 的配額

AWS 您的帳戶具有每個 AWS 服務的預設配額，先前稱為限制。除非另有說明，否則每個配額都是區域特定的。您可以請求提高某些配額，而其他配額無法提高。

若要檢視 Application Load Balancer 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 Elastic Load Balancing。您也可以對 Elastic Load Balancing 使用 describe-account-limits (AWS CLI) 命令。

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，請提交提高服務配額的請求。

配額

- 負載平衡器
- 目標群組
- Rules
- 信任存放區
- 憑證
- HTTP 標頭
- Load Balancer容量單位

負載平衡器

AWS 您的帳戶具有與 Application Load Balancer 相關的下列配額。

Name	預設	可調整
每個區域的 Application Load Balancer	50	是
每個 Application Load Balancer 的憑證 (不含預設憑證)	25	是
每個 Application Load Balancer 的接聽程式	50	是
每個 Application Load Balancer 每個動作的目標群組	5	否

Name	預設	可調整
每個 Application Load Balancer 的目標群組	100	否
每個 Application Load Balancer 的目標	1,000	是

目標群組

下列配額適用於目標群組。

Name	預設	可調整
每個區域的目標群組	3,000 *	是
每個區域每個目標群組的目標數 (執行個體或 IP 地址)	1,000	是
每個區域每個目標群組的目標數 (Lambda 函數)	1	否
每個目標群組的負載平衡器	1	否

* 此配額由 Application Load Balancer 和 Network Load Balancer 共用。

Rules

下列配額適用於規則。

Name	預設	可調整
每個 Application Load Balancer 的規則 (不含預設規則)	100	是
每個規則的條件值	5	否
每個規則的條件萬用字元	6	否
每個規則的比對評估次數	5	否

信任存放區

下列配額適用於信任存放區。

Name	預設	可調整
每個帳戶的信任存放區	20	是
每個負載平衡器在驗證模式下使用 mTLS 的接聽程式數量。	2	否

憑證

下列配額適用於憑證，包括公告 CA 憑證名稱和憑證撤銷清單。

Name	預設	可調整
CA 憑證大小	16 KB	否
每個信任存放區的 CA 憑證	25	是
每個信任存放區的 CA 憑證主體大小	10,000	是
憑證鏈深度上限	4	否
每個信任存放區的撤銷項目	500,000	是
撤銷清單檔案大小	50 MB	否
每個信任存放區的撤銷清單	30	是
TLS 訊息大小	64 K	否

HTTP 標頭

HTTP 標頭的大小限制如下。

Name	預設	可調整
請求行	16 K	否
單一標頭	16 K	否
整個回應標頭	32 K	否
整個請求標頭	64 K	否

Load Balancer容量單位

下列配額適用於Load Balancer容量單位 (LCU)。

Name	預設	可調整
每個 Application Load Balancer LCUs) Application Load Balancer	15,000	是
每個區域的預留 Application Load Balancer 容量單位 (LCU)	0	是

Application Load Balancer 的文件歷史記錄

下表說明 Application Load Balancer 各版本。

變更	描述	日期
存取字符驗證	此版本新增對負載平衡器的支援，以驗證用戶端為安全 service-to-service(S2S) 或 machine-to-machine(JWT)。	2025 年 11 月 21 日
轉換	此版本新增支援，可在負載平衡器將流量路由至目標之前，轉換傳入請求的主機標頭和 URLs。	2025 年 10 月 15 日
存取日誌和連線日誌的儲存貯體政策	在此版本之前，您使用的儲存貯體政策取決於 區域是否在 2022 年 8 月之前或之後可用。在此版本中，所有 區域都支援較新的儲存貯體政策。請注意，仍然支援舊版儲存貯體政策。	2025 年 9 月 10 日
HTTP 標頭修改	此版本新增了對所有回應代碼 HTTP 標頭修改的支援。先前，此功能僅限於回應代碼 2xx 和 3xx。	2025 年 2 月 28 日
容量單位保留	此版本新增了為負載平衡器設定最小容量的支援。	2024 年 11 月 20 日
資源映射	此版本新增支援，以視覺化格式檢視您的負載平衡器資源和關係。	2024 年 3 月 8 日

按一下 WAF	此版本新增了在與一鍵式整合時設定負載平衡器行為的支援 AWS WAF。	2024 年 2 月 6 日
相互 TLS	此版本新增了對交互 TLS 身分驗證的支援。	2023 年 11 月 26 日
自動目標權重	此版本新增了對自動目標權重演算法的支援。	2023 年 11 月 26 日
FIPS 140-3 TLS 終止	此版本新增了在終止 TLS 連線時使用 FIPS 140-3 crypotographic 模組的安全政策。	2023 年 11 月 20 日
使用 IPv6 註冊目標	此版本新增支援，以在 IPv6 處理時將執行個體註冊為目標。	2023 年 10 月 2 日
支援 TLS 1.3 的安全政策	此版本新增了對 TLS 1.3 預先定義安全政策的支援。	2023 年 3 月 22 日
區域轉移	此版本新增支援，透過與 的整合，將流量從單一受損的可用區域路由離開 Amazon 應用程式復原控制器 (ARC)。	2022 年 11 月 28 日
關閉跨區域負載平衡	此版本新增了關閉跨區域負載平衡的支援。	2022 年 11 月 28 日
目標群組運作狀態	此版本新增的支援，可讓您設定必須處於運作狀態良好之目標的最小計數或百分比，以及不符合閾值時負載平衡器採取的動作。	2022 年 11 月 28 日
跨區域負載平衡	此版本新增了在目標群組層級設定跨區域負載平衡的支援。	2022 年 11 月 17 日

IPv6 目標群組	此版本新增支援，可讓您為 Application Load Balancer 設定 IPv6 目標群組。	2021 年 11 月 23 日
IPv6 內部負載平衡器	此版本新增支援，可讓您為 Application Load Balancer 設定 IPv6 目標群組。	2021 年 11 月 23 日
AWS PrivateLink 和靜態 IP 地址	此版本透過將流量直接從 Network Load Balancer 轉送至 Application Load Balancer，新增使用 AWS PrivateLink 和公開靜態 IP 地址的支援。	2021 年 9 月 27 日
用戶端連接埠保留	此版本新增屬性，可保留用戶端用來連線到負載平衡器的來源連接埠。	2021 年 7 月 29 日
TLS 標頭	此版本新增 屬性，指出包含交涉 TLS 版本和密碼套件相關資訊的 TLS 標頭，會在傳送至目標之前新增至用戶端請求。	2021 年 7 月 21 日
其他 ACM 憑證	此版本支援具有 2048、3072 和 4096 位元金鑰長度的 RSA 憑證，以及所有 ECDSA 憑證。	2021 年 7 月 14 日
應用程式型粘性	此版本新增應用程式型 Cookie，以支援負載平衡器的粘性會話。	2021 年 2 月 8 日
支援 TLS 1.2 版之 FS 的安全政策	此版本新增支援 TLS 1.2 版向前保密 (FS) 的安全政策。	2020 年 11 月 24 日
WAF 故障開啟支援	此版本新增了在與 整合時設定負載平衡器行為的支援 AWS WAF。	2020 年 11 月 13 日

gRPC 和 HTTP/2 支援	此版本新增對 gRPC 工作負載和端對端 HTTP/2 的支援。	2020 年 10 月 29 日
Outpost 支援	您可以在 上佈建 Application Load Balancer AWS Outposts。	2023 年 9 月 8 日
去同步緩解模式	此版本新增對非同步緩和模式的支援。	2020 年 8 月 17 日
最少未完成的請求	此版本增加對最少未完成請求演算法的支援。	2019 年 11 月 25 日
加權目標群組	此版本增加對多個目標群組轉送動作的支援。請求會根據您為每個目標群組指定的權重分配至這些目標群組。	2019 年 11 月 19 日
New attribute (新建屬性)	此版本新增對路由 http.drop_invalid_header_fields.enabled 屬性的支援。	2019 年 11 月 15 日
FS 的安全政策	此版本新增了對三個額外預先定義轉送私密安全政策的支援。	2019 年 10 月 8 日
進階請求路由	此版本對接聽程式規則增加支援其他條件類型。	2019 年 3 月 27 日
Lambda 函數作為目標	此版本增加將 IP 函數註冊為目標的支援。	2018 年 11 月 29 日
重新導向動作	此版本增加對負載平衡器將請求重新導向不同 URL 的支援。	2018 年 7 月 25 日
固定回應動作	此版本增加對負載平衡器傳回自訂 HTTP 回應的支援。	2018 年 7 月 25 日

FS 和 TLS 1.2 的安全政策	此版本增加對額外兩個預先定義安全政策的支援。	2018 年 6 月 6 日
使用者身分驗證	此版本增加對負載平衡器在路由傳送請求之前，使用企業或社交身分來驗證應用程式使用者的支援。	2018 年 5 月 30 日
資源層級許可	此版本增加對資源層級許可和標記條件金鑰的支援。	2018 年 5 月 10 日
慢速啟動模式	此版本增加對慢速啟動模式的支援，該模式可在負載平衡器預備時，逐漸增加負載平衡器傳送到新註冊目標的請求量。	2018 年 3 月 24 日
SNI 支援	此版本增加對伺服器名稱指示 (SNI) 的支援。	2017 年 10 月 10 日
IP 地址即目標	此版本新增了支援註冊 IP 地址做為目標。	2017 年 8 月 31 日
主機型路由	此版本增加根據主機標頭中主機名稱來路由請求的支援。	2017 年 4 月 5 日
TLS 1.1 和 TLS 1.2 的安全政策	此版本增加 TLS 1.1 和 TLS 1.2 的安全政策。	2017 年 2 月 6 日
IPv6 支援	此版本增加對 IPv6 地址的支援。	2017 年 1 月 25 日
請求追蹤	此版本增加對請求追蹤的支援。	2016 年 11 月 22 日
TargetResponseTime 指標的百分位數支援	此版本新增對 Amazon CloudWatch 所支援之新百分位數統計資料的支援。	2016 年 11 月 17 日

新的負載平衡器類型	此 Elastic Load Balancing 版本推出 Application Load Balancer。	2016 年 8 月 11 日

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。